Multi-Field Visualization

Vom Fachbereich Informatik
der Technischen Universität Kaiserslautern
zur Erlangung des akademischen Grades
Doktor der Naturwissenschaften
(Dr. rer. nat.)
genehmigte

Dissertation

von

Harald Obermaier

Datum der wissenschaftlichen Aussprache: 9. Februar 2011

Dekan: Prof. Dr. Karsten Berns
Prüfungskommission:
    Vorsitz: Prof. Dr. Klaus Madlener
    Erster Berichterstatter: Prof. Dr. Hans Hagen
    Zweiter Berichterstatter: Prof. Dr. Martin Hering-Bertram
    Dritter Berichterstatter: Prof. Dr. Bernd Hamann

D 386

delivered by
EUROGRAPHICS DIGITAL LIBRARY
www.eg.org diglib.eg.org

Acknowledgments

Throughout my studies and the preparation of this thesis I had the honor of being supported by advisors, colleagues, and friends in countless ways and without whose help the present work would not have been possible.

For their highly valuable advise during my PhD studies I want to thank Hans Hagen and Martin Hering-Bertram, who were always open for questions and fruitful discussions. I would like to thank everyone who reviewed and commented on my thesis with feedback that I highly appreciated. Furthermore, I wish to thank all members of the International Research Training Group of the University of Kaiserslautern (IRTG 1131) and of the Transport-Processes group of the Fraunhofer ITWM for creating and providing the best research climate I could have hoped for and for enabling me to pursuit successful and international doctorate studies.

My thanks go to our fellow researchers at the University of California, Davis and the University of Utah for hosting, advising, and supporting me during my international research stays in California and Utah. Without them and my colleagues from the IRTG, these stays would not have been as productive and enjoyable as they truly were. For organizing, and helping out on countless occasions and matters during my stays abroad and in Germany, I wish to give special thanks to Mady and Inga.

I want to thank my family: my parents and my brothers for their never-ending support during all of my life and my studies. For bearing with me during times of long work and everything else she has done for me I want to thank Julia.

Abstract

Modern science utilizes advanced measurement and simulation techniques to analyze phenomena from fields such as medicine, physics, or mechanics. The data produced by application of these techniques takes the form of multi-dimensional functions or fields, which have to be processed in order to provide meaningful parts of the data to domain experts. Definition and implementation of such processing techniques with the goal to produce visual representations of portions of the data are topic of research in scientific visualization or multi-field visualization in the case of multiple fields.

In this thesis, we contribute novel feature extraction and visualization techniques that are able to convey data from multiple fields created by scientific simulations or measurements. Furthermore, our scalar-, vector-, and tensor field processing techniques contribute to scattered field processing in general and introduce novel ways of analyzing and processing tensorial quantities such as strain and displacement in flow fields, providing insights into field topology.

We introduce novel mesh-free extraction techniques for visualization of complex-valued scalar fields in acoustics that aid in understanding wave topology in low frequency sound simulations. The resulting structures represent regions with locally minimal sound amplitude and convey wave node evolution and sound cancellation in time-varying sound pressure fields, which is considered an important feature in acoustics design.

Furthermore, methods for flow field feature extraction are presented that facilitate analysis of velocity and strain field properties by visualizing deformation of infinitesimal Lagrangian particles and macroscopic deformation of surfaces and volumes in flow. The resulting adaptive manifolds are used to perform flow field segmentation which supports multi-field visualization by selective visualization of scalar flow quantities.

The effects of continuum displacement in scattered moment tensor fields can be studied by a novel method for multi-field visualization presented in this thesis. The visualization method demonstrates the benefit of clustering and separate views for the visualization of multiple fields.

Zusammenfassung

Wissenschaftliche Messungen und Simulationen erzeugen Daten, mit deren Hilfe komplexe physikalische Zusammenhänge und Phänomene modelliert und analysiert werden können. Die hierdurch enstandene Menge an Felddaten kann ohne Abstraktions- und Aufbereitungsmaßnahmen nur selten direkt interpretiert werden. Ziel der Scientific Visualization ist es, eine hinreichende Abstraktionsmittel durch die Definition und Extraktion von aussagekräftigen Datenmerkmalen zur Verfügung zu stellen und diese angemessen visuell darzustellen. Visualisierung für multiple Felder wird als Multi-Field Visualization bezeichnet.

In dieser Dissertation entwickeln wir neue Techniken zur Merkmalsextraktion und Visualisierung mit Anwendung im Kontext der Multi-Field Visualisierung. Zwar sind die vorgestellten Techniken in der Regel unabhängig von der vorhandenen Nachbarschaftsstruktur der Daten, dennoch betonen wir die Anwendbarkeit der entwickelten Methoden in gitterfreien Datensätzen. Eine weitere mathematische Gemeinsamkeit der Methoden besteht in der neuartigen Analyse und Einbindung von Deformationsdaten in den Extraktions- und Visualisierungprozess.

Wir entwerfen eine neue gitterfreie Methode zur Extraktion generalisierter Extrema in dreidimensionalen komplexwertigen Skalarfeldern. Diese Skalarfelder sind das Resultat von niedrigfrequenten Akustiksimulationen, in denen Topologie und extremale Merkmale in Wellenstrukturen von großer Bedeutung sind. Unsere Methode erlaubt es, Wellenknoten und Minimalamplituden in stehenden und bewegten Wellen in komplexwertigen Schalldruckfeldern zu extrahieren und visualisieren, was von zentralem Interesse im Gebiet des Akustikdesigns ist.

Desweiteren untersuchen wir Deformationen in Vektorfeldern. Die von uns präsentierten Techniken zur Einbindung dieser Deformationstensoren in die integrale Vektorfeldvisualierung erlauben die Visualisierung vektorieller und tensorieller Vektorfeldgrößen und liefern Informationen über Divergenz, Konvergenz und Mischverhalten der untersuchten Strömungen. Eine vorgestellte Erweiterung dieses Konzepts auf volumetrische Datensätze aus der Geophysik ermöglicht die Segmentierung und selektive Visualisierung von Strömungsvorgängen in der Erdkruste. Zusätzlich stellen wir neue Ansätze zur adaptiven Generierung und Visualisierung makroskopischer Deformationen von Gebiets- und Flächenstrukturen vor.

Als verwandtes physikalisches Verhalten analysieren wir Verschiebungsdaten in der Form von Momententensoren. Zur aussagekräftigen Darstellung der Messungsdaten entwickeln wir Multi-Field Visualiserungtechniken, die auf der Verwendung gekoppelter Datenansichten basieren und die interaktive Analysen im dreidimensionalen und projektiven Raum ermöglichen.

Contents

1. Introduction **1**

2. Scientific Visualization **5**

- 2.1. Geometry and Topology 6
- 2.2. Field Visualization 7
  - 2.2.1. Scalar Fields 8
    - 2.2.1.1. Application Areas 8
    - 2.2.1.2. Features of Scalar Fields 9
    - 2.2.1.3. Scalar Field Topology 11
    - 2.2.1.4. Scalar Field Visualization Techniques 12
  - 2.2.2. Vector Fields 13
    - 2.2.2.1. Application Areas 13
    - 2.2.2.2. Features of Vector Fields 13
    - 2.2.2.3. Vector Field Topology 14
    - 2.2.2.4. Vector Field Visualization Techniques 15
  - 2.2.3. Second Order Tensor Fields 16
    - 2.2.3.1. Application Areas 16
    - 2.2.3.2. Features of Tensor Fields 17
    - 2.2.3.3. Tensor Field Topology 17
    - 2.2.3.4. Tensor Field Visualization Techniques 18
  - 2.2.4. Multi-Field 18
    - 2.2.4.1. Application Areas 18
    - 2.2.4.2. Multi-Field Features 19
    - 2.2.4.3. Multi-Field Visualization 19
- 2.3. Field Representation 19
- 2.4. Field Reconstruction 20

3. Mathematical Definitions, Properties, and Methods **23**

- 3.1. Notation and Prerequisites 23
- 3.2. Linear Algebra and Vector Calculus 23
  - 3.2.1. Fields, Operators 24
  - 3.2.2. Numerical Methods 27
    - 3.2.2.1. Numerical Differentiation 27
    - 3.2.2.2. Tensor Decompositions 27

3.3. Continuum Mechanics 27
    3.3.1. Fluid Mechanics 28
    3.3.2. Solid Mechanics 29
        3.3.2.1. Deformation 29
    3.3.3. Moments 31
    3.3.4. Numerical Methods 32
3.4. Differential Geometry 32
    3.4.1. Integral Features 32
    3.4.2. Numerical Methods 34
3.5. Field Topology 35
    3.5.1. Scalar Field Topology 35
    3.5.2. Vector Field Topology 36
        3.5.2.1. Stationary Vector Fields 37
        3.5.2.2. Instationary Vector Fields 38
    3.5.3. Tensor Fields 39
    3.5.4. Numerical Methods 39

4. Complex Valued Scalar Fields - Mesh-Free Valley Surfaces 41
4.1. Related Work 42
    4.1.1. Crease Extraction 42
    4.1.2. Sound Visualization 44
4.2. Crease Surface Theory 45
4.3. Valley Surface Construction 46
    4.3.1. Extraction 46
        4.3.1.1. Pre-Processing of the Data Set 46
        4.3.1.2. Initialization 47
        4.3.1.3. Point Seeding 47
        4.3.1.4. Converging to the Valley Surface 48
        4.3.1.5. Merging of Points 50
        4.3.1.6. Independent Surfaces 50
        4.3.1.7. Progressive Surface Generation 50
    4.3.2. Non-Manifold Regions 51
        4.3.2.1. Surface Branching 52
    4.3.3. Triangulation 54
4.4. Application 55
4.5. Results 56
    4.5.1. Application to Sound Visualization 56
    4.5.2. Evaluation 60
4.6. Summary and Discussion 63

5. Strain in Flow Fields **67**
5.1. Strain Advection 68
5.1.1. Computation 68
5.1.1.1. Strain Accumulation 68
5.1.1.2. Multiplication Scheme 70
5.1.2. Strain Along Streaklines 72
5.2. Related Work 73
5.2.1. Visualization 73
5.2.2. Geophysics 74
5.3. Strain Field Analysis 74
5.3.1. Motivation 74
5.3.2. Strain in Geophysical Flow Data 75
5.3.2.1. Background 75
5.3.2.2. Data Format 76
5.3.3. Strain Field Computation 77
5.3.3.1. Uniform Discretization 78
5.3.3.2. Pathline Integration 79
5.3.3.3. Orientation Field Computation 79
5.3.3.4. Stationary Strain Fields 80
5.3.4. Strain Field Segmentation 80
5.3.4.1. Orientation Segmentation 80
5.3.4.2. Alignment Segmentation 81
5.4. Visualization 82
5.4.1. Trace Visualization 82
5.4.1.1. Finite Integration Lengths 83
5.4.2. Strain Field Visualization 83
5.4.2.1. Strain Lines 84
5.4.2.2. Multi-Volume Visualization 84
5.4.2.3. Strain Histograms 87
5.5. Results and Application 89
5.5.1. Instantaneous Flow 90
5.5.2. Time-Varying Flow 92
5.6. Summary and Discussion 95

6. Unsteady Flow Segmentation **99**
6.1. Related Work 100
6.2. Particle Advection 101
6.3. Adaptivity 102
6.3.1. Insertion of Particles 103
6.3.2. Deletion of Particles 103
6.4. Adaptive Streak Areas 103
6.4.1. Streakline Integration 104
6.4.1.1. Refinement 104

6.4.1.2. Boundary Integration 105
6.4.1.3. Field Segmentation 106
6.5. Adaptive Streak Surfaces 106
6.5.1. Streak Surfaces 106
6.5.2. Surface Generation 107
6.5.2.1. Algorithm Outline 107
6.5.2.2. Particle Birth 107
6.5.2.3. Particle Advection 108
6.5.2.4. Delaunay Meshing 109
6.5.2.5. Curvature Approximation 111
6.5.2.6. Particle Insertion 112
6.6. Results 114
6.7. Summary and Discussion 120

7. Tensor Fields - Scattered Moment Tensor Data 125
7.1. Related Work 126
7.2. The Moment Tensor 127
7.2.1. Fault Representation 127
7.2.2. Moment Tensor Decomposition 128
7.2.3. Moment Tensor Properties 129
7.3. Moment Tensor Glyphs 129
7.3.1. Polarity Glyphs 130
7.3.1.1. Visualization 131
7.3.2. Slip Geometry 131
7.3.2.1. Visualization 132
7.4. Clustering and Averaging 132
7.4.1. Similarity Measures 133
7.4.1.1. Double-Couple Rotation 133
7.4.1.2. Full Tensor Similarity 134
7.4.2. Clustering 135
7.4.2.1. Distance-Quality-Threshold Clustering 135
7.4.3. Tensor Averaging 136
7.4.3.1. Cluster Visualization 137
7.5. Interactive Visualization 139
7.5.1. Stereonet Display 139
7.5.2. Interaction 139
7.6. Results 140
7.7. Summary and Discussion 144

8. Conclusions 145

A. Moving Least Squares 147
A.1. Definition 147

A.2. Properties 147
A.2.1. Accuracy 148
A.2.2. Complexity 148
A.2.3. Derivatives 149
A.3. Challenges 149

B. Voxelization 151
B.1. GPU Voxelization 151
B.2. CPU Voxelization 151
B.3. Volume Voxelization 152

References 154

List of Figures 173

Curriculum Vitae 176

1. Introduction

Scientific simulation and measurement techniques produce increasing amounts of numerical data every day. To put a meaning to these sets of numbers and provide domain experts with an intuitive visual understanding of their data sets is the central motivation for the discipline of scientific visualization.

Data obtained from modern simulation and measurement techniques includes large and complex data sets, covering diverse areas such as Computational Fluid Dynamics (CFD) in mathematics and physics, (DT)-MRI data in medicine, and stress or moment data in geology. Output of common scientific simulations is usually not limited to a single, distinct data type, but consists of a number of mappings into spaces of different dimensions. Examples of such fields are scalar fields representing temperature or pressure and vector fields describing the velocity of fluids. Additional related higher or lower dimensional fields may be derived by mathematical transformations or combinations of the simulation output. Thus, common data dimensions reach from single scalar valued fields over n-dimensional vector fields to high-order tensor fields and combinations thereof and confront experts with the problem of interpreting data of increasing resolution, size and number. The challenge of visualizing data from multiple fields comes from the need to present experts with a meaningful portion of this overwhelming amount of data. One way to meet these requirements is the definition of new features and extraction methods for different types of fields and domains, especially in the context of new measurement and simulation techniques that sample the data domain irregularly and in a mesh-free manner.

Central goal of feature definitions and new visualization methods is to compute visual representations that allow an in-depth analysis of the simulation output. As simulation processes often yield a black box behavior, new visualization techniques can be used for a thorough analysis of underlying processes and parameter identification.

Depending on the questions asked by domain experts, different portions of a field's data are relevant to the answer. Therefore, feature definitions can be viewed as mathematical answers to abstract information descriptions. An example of an interesting property in the context of vector field visualization is the amount and quality of mixing performed by the underlying flow. Appropriate vector field features providing an insight into this property try to visually separate distinct flow regions into regions of homogeneous flow, leading to the challenge of constructing separation geometry or other visual metaphors that are both reasonably fast to compute and accurate in their representation of the analyzed mixing property. A common way to balance between accuracy and computation speed is the use of spatially adaptive structures that dynamically adjust resolution or structural complexity based on the local feature size or desired reconstruction quality. This notion of adaptivity is especially important in irregularly sampled data sets or scattered point sets in general, where local variation in sample density can be large. Important loci in scalar fields include regions where data values are equal or locally extreme, whereas tensor field analysis is often concerned with the extraction of degenerate manifolds. An important observation about scalar, vector and tensor fields is the ability to gain information about these high order fields by reducing them to a characteristic lower order representation and vice versa. Therefore, not only the extraction of features in one field type is desired, but the benefits to the analysis of other field types and accompanying feature analogies are of great significance to visualization of multiple fields.

State-of-the-art feature extraction techniques are mostly concerned with regularly sampled data or fields with a predefined neighborhood relationship between adjacent sample points. These data sets leave room for new adaptive feature extraction and reconstruction techniques that bear the possibility to produce results in manually or automatically defined levels of detail, as no neighborhood size is pre-specified. Furthermore, analysis of related fields and feature definitions is a promising topic in the area of multi-field visualization, as it not only allows analysis of independent or related fields but facilitates the understanding of fields by incorporating derived quantities into the visualization process.

In this dissertation we make the following contributions: We facilitate the analysis of three-dimensional wave node structures in simulated acoustics by extraction of minimum structures in complex-valued scalar fields [OMD$^+$] in Chapter 4. Our method generates adaptive (non-manifold) meshes by a new grid-less surface growing technique, based on analysis of the Hessian matrix of the amplitude-field. The developed method is applied to low frequency sound simulations and allows (interactive) and time-varying visualization of wave node structures in three-dimensional space. Results show amplitude and phase fields visualized as color coded surface structures in real-world and synthetic data sets, demonstrating the usefulness of visualization of pressure minima for acoustics engineering.

The importance of tensorial measures in fields, that is indicated in Chapter 4 is further emphasized by our work on strain in flow fields [OHBKH09b, OBH$^+$] in Chapter 5, where we develop methods to compute and visualize flow induced strain along integral flow features. The resulting concepts enhance integral flow visualization techniques by incorporating a neighborhood-aware measure into the extraction and visualization process. Further, we devise techniques for strain visualization in geophysical data sets. Our numerical results combining flow and strain information show how strain measures in flow help understand mixing and faulting properties of various flow simulation data sets.

Challenges in integral flow field visualization as presented in Chapter 5 include the extraction of integral flow geometry that samples the underlying flow field in a sufficiently dense and reliable manner. To overcome limitations of static integral flow geometry, we introduce novel adaptive integral flow feature extraction algorithms that are designed to work in scattered two- and three-dimensional data sets [OHBKH09b, OHBKH09a] in Chapter 6. A benefit of the resulting feature geometry is its use as segmentation geometry for multi-field visualization as pointed out in the results section along with other advantages of our adaptive extraction techniques.

We enhance state-of-the-art visualization techniques for moment tensor analysis and visualization [OBHHB11] in Chapter 7. The analyzed tensors contain slip information related to shear strain that can be observed in flow fields. We propose tensor clustering and visualization techniques that aid domain experts in understanding the complex relationships between individual tensors in a scattered data set. The presented results combine orientation based visualization and tensor glyph display and facilitate interactive exploration and analysis of tensor clusters.

In summary, the scientific contributions of this thesis are organized as follows: Chapter 2 provides an background information on feature-based scientific visualization of scalar-, vector-, and tensor fields from an application point-of-view. Following this general introduction into the state-of-the-art, Chapter 3 defines the necessary mathematical concepts, properties, and methods that are relevant to the scope of this work and are used frequently throughout this thesis. Ordered by increasing dimension of the governing field, Chapters 4 (scalar fields), 5,6 (flow fields), and 7 (tensor fields) develop our novel methods by introducing specific related work, methods and algorithms as well as results. This thesis is concluded in Chapter 8. We give details about the scattered data approximation techniques and voxelization methods used throughout this thesis in Appendix A and B.

2. Scientific Visualization

Scientific Visualization is a discipline that aims at creating insightful visual representations of a given "set of numbers" obtained from scientific measurements or simulations. In contrast, visualization of meta-data or otherwise highly abstract/non-spatial data is commonly termed *Information Visualization.*

The focus of this thesis lies almost exclusively on scientific visualization and the branch of feature-based visualization in particular. *Feature-Based Visualization* methods have the goal to create visual abstractions that convey one or more important properties of the given data as identified by domain experts and stands in contrast to direct visualization techniques that refrain from complex processing of the field itself and visualize local properties of the data directly.

Given an abstract question (*"Where are discontinuities in the data set?"*) or mathematical feature definition (*"Where do first derivatives vanish?"*), creating an insightful feature-based visualization means to transform this definition into a valid mathematical context, define feature extraction and processing techniques, and finally produce a suitable visual representation of these meaningful parts of the data set. In this work we therefore regard the visualization process as a three-step pipeline as shown in Figure 2.1 and detailed in the following.

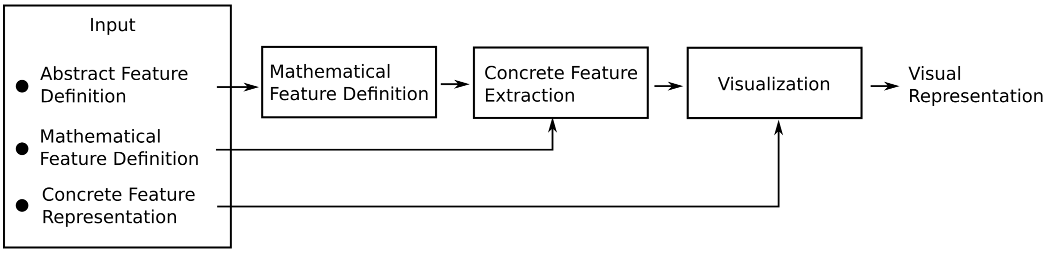

Input
Abstract Feature Definition
Mathematical Feature Definition
Concrete Feature Representation
Mathematical Feature Definition
Concrete Feature Extraction
Visualization
Visual Representation

Figure 2.1.: Three-step pipeline of feature-based visualization and possible input types.

1. *Feature Definition*: Given an abstract feature definition or question, this step identifies mathematical properties and provides the necessary mathematical feature definition in order for the extraction process to be well-defined. In general, the act of defining a feature corresponds to an information reduction and densification step that serves to highlight contextually important parts of the data set while hiding redundant information.

2. *Extraction*: The concrete representation of the feature corresponds to the solution of the mathematical description given in step one. The feature extraction step is concerned with the definition of analytic or numerical methods that allow explicit construction of a (geometric) feature representation.

3. *Visualization*: This step provides the user with an insightful visual depiction of the (geometric) feature representation and is concerned with avoiding ambiguities or occlusion and solves general problems related to human perception.

Depending on the form of the given input, i.e. whether an initial mathematical feature definition or representation is available a priori, the focus of the development of a concrete visualization technique shifts substantially. As implied by the role of domain knowledge in the abstract feature definition process, features are highly application-specific. However, over the years of visualization research, a set of feature types has proven to be especially helpful in answering a variety of application questions.

Before giving an overview of these feature definitions for scalar-, vector-, and tensor fields in the following sections, we first provide an introduction to the concept of topology and topological features. We then present the relevant state-of-the-art visualization techniques and concepts, and motivate important feature and topology definitions for scalar-, vector- and tensor fields. These sections aim at providing a high-level description of field visualization with focus on application specific relevance. Detailed treatment of related work and feature definitions are given in Chapter 4 and following.

2.1. Geometry and Topology

A mathematical concept with major impact on feature definitions in visualization is given by the field of topology. *Geometry* and *Topology* are closely related concepts in mathematics. While geometry is concerned with concrete properties of space such as the length, shape, size and location of objects, topology as an abstraction of classic geometry focuses on certain relative properties of space that are invariant under transformations such as scaling and deformation. A simple example of a question that can be answered using topological considerations and that is independent of concrete shapes or lengths is the question if and how things are "connected"[RS05]. Therefore, topological features are often regarded as providing an abstract view of the connectivity or "adjacency" of space. While mathematics defines topology on arbitrary sets, we limit the notion of topology to concrete geometric entities in this thesis.

Similarity or equivalence in classic geometry is usually defined by the use of element-based distance measures. Objects are regarded as similar if their difference in shape with respect to a pre-defined metric falls below a certain threshold. The notion of equivalence with respect to topology requires fundamentally different concepts and leads to the definition of topological spaces and homeomorphisms: In the case of this thesis, it is sufficient to note that euclidean space $\mathbb{R}^n$, as all *metric spaces*, is a *topological space*. For abstract definitions of topological and metric spaces we refer to the literature [RS05]. Two topological spaces X, Y are *topologically equivalent*, if there exists a continuous one-to-one mapping $f : X \to Y$ with continuous inverse f^{-1}. Such a function is called a *homeomorphism*, and X,Y are *homeomorphic* if such f exists. With this definition, we can safely answer questions about similarity in object connectivity and distinguish between pure geometry and topology of an object, as seen in Figure 2.2. *Topological features* in fields are features, that give an impression of connectivity in the field and remain structurally stable for small perturbations of the field.

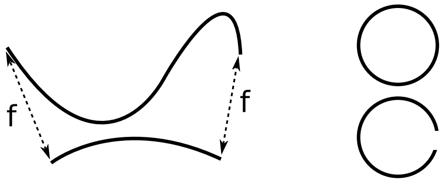


Figure 2.2.: Left: Two shapes that are topologically equivalent despite showing dissimilar geometry. Right: Two topologically inequivalent shapes that might be considered similar with respect to geometry.

Topology definitions on a given space can be used to perform simplifications of the domain by identification of topologically relevant features and connected components. Thus, data reduction and densification of visual information of complex structures is achieved by limiting the visual representation to the topological skeleton described by these connected components. As these topological features and their graphs represent an important area of feature-based visualization, the following sections provide details about general features as well as field topology analysis.

2.2. Field Visualization

Some areas of scientific visualization are interested in depicting direct spatial properties of the obtained data set such as shape and orientation, while visualization of fields is mainly concerned with analytical, topological and numerical properties of the represented function. For this reason, visualization of fields is closely related to the area of mathematical analysis, as emphasized by the mathematical definitions and concepts given in Chapter 3.

Goal of the following sections is to provide an abstract overview of the state-of-the-art in related scientific field visualization techniques and to establish the relevant high-level scientific context of this thesis. In the following we differentiate fields according to the dimension of their output. Each section gives a brief insight into application areas of the respective field, (topological) feature motivation, and visualization techniques. A more technical definition of the field and feature properties is given in Chapter 3.

2.2.1. Scalar Fields

Scalar fields are the lowest dimensional form of general tensor fields, holding one real or complex number per data point. In two-dimensional space, regular scalar fields may be interpreted as (discrete) gray-scale images or regular height fields, as seen in Figure 2.3. Consequently, the area of scalar field visualization is closely related to the field of image processing [Ban08].

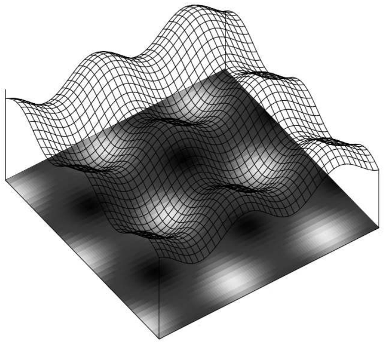


Figure 2.3.: Scalar field shown as gray-scale image and height-field.

2.2.1.1. Application Areas

Scientific data is usually produced either by real-world measurements or (numerical) simulations, or is of purely synthetic nature. In the following, we present application areas whose scalar-valued output has a strong impact on the visualization community and are relevant to the remainder of this thesis.

A prominent source of scalar fields from real-world measurements are imaging techniques in areas such as medicine with its *Computed Tomography* (CT) and *Magnetic Resonance Imaging* (MRI) [UH91, LL00] scanners which generate (volumetric) density fields of the scanned subject. While the slices of the resulting scalar fields representing individual two-dimensional scans can be viewed individually with standard image viewers and parameters of medical scanners can be changed to highlight different features of the scanned object, the need for automatic or assisted detection of (three-dimensional) anomalies calls for advanced feature extraction and visualization techniques.

While scalar fields are by-products of a wide range of simulation techniques [Bat67], several physics simulations focus especially on scalar-valued output such as temperature and field strength. The particular class of sound simulations [Sab22] is of special interest for this thesis, as seen in Chapter 4. General *acoustics simulations* are concerned with the imitation of wave behavior in matter. The sub-field of sound simulations concentrates on wave properties in gaseous matter and implementations of these simulations serve as prediction tools for acoustics engineering and design.

Given a specific room geometry, the central challenge of room acoustics simulations is the construction of an impulse response filter for each pair of speaker and listener positions. To fully describe the impulse response of a room, these filters have to be reconstructed for a range of frequencies, as different materials present in the room have different sound absorption and reflection properties. In the course of these computations, a three-dimensional complex valued scalar field is produced that keeps sound amplitude and phase values for a given input frequency at densely sampled listener positions [Dei08]. Prominent features of these output fields are of interest for acoustics engineering, as they allow parameter optimization during the design process of acoustic environments.

2.2.1.2. Features of Scalar Fields

In medical applications, feature definitions generally target at the detection of anomalies such as fractures or tumors [Ban08]. In practice, such features can be found by performing a manual search for highlights, edges or discontinuities in direct two-dimensional visualizations of slices of CT or MRI scans. For the processing of volumes represented as stacks of medical scans, manual inspection however requires cumbersome slicing or region-of-interest selection to avoid occlusion. If the feature can be described mathematically, this process can be supported or automated by feature based visualization.

While discontinuities are of great interest in simulation data processing and simulation verification as well, questions from domain experts often regard analytic and numerical properties or the general structure and topology of the field [Ban08].

Fortunately analysis of (one-dimensional) scalar fields is a mathematically well-known topic and has been studied extensively for centuries. Classic one-dimensional function analysis yields a number of mathematical feature definitions (see Figure 2.4), including features such as:

1. (local) extrema/critical points
2. plateaus
3. inflection points
4. discontinuities

These definitions generalize to higher-dimensional scalar fields and can be used to answer a broad field of application questions. In medicine, discontinuities in the density field can indicate broken bones, highlights may imply tumors, and so on [Ban08].

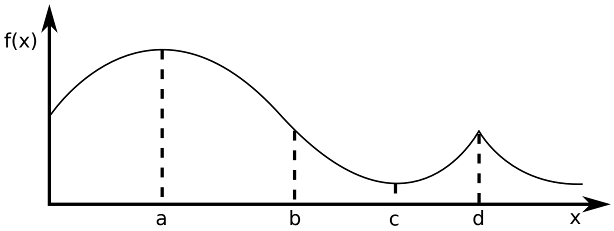


Figure 2.4.: One-dimensional scalar-valued function with inner maximum (a), minimum (c), inflection point (b) and discontinuous derivative (d).

In addition to these features, higher-dimensional scalar field analysis requires features that allow contouring or shape recognition. Typical feature definitions for scalar field contouring highlight regions of constant function value, so called *level sets* L (or *isocontours/isosurfaces*)

$$L_c = \{x \in \mathbb{R}^n | f(x) = c\}, \tag{2.1}$$

interval regions I

$$I_{ab} = \{x \in \mathbb{R}^n | a < f(x) < b\}, \tag{2.2}$$

or locations that correspond to a certain behavior of derivatives:

$$G_{ab} = \{x \in \mathbb{R}^n | a < \|\nabla f\| < b\}. \tag{2.3}$$

These outlines can be used for automatic detection of bone shapes, as shown in Figure 2.5.

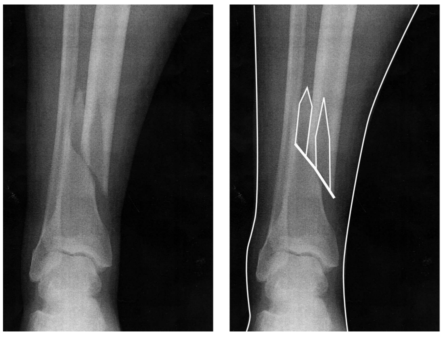


Figure 2.5.: X-Ray image of a broken bone. Right: Isocontour represents leg outline, discontinuity shows bone fracture.

If a scalar field is interpreted as height-field, an interesting question is the shape and location of valleys or height-ridges [Ebe96] which is studied in greater detail in Chapter 4.

Other interesting features of scalar fields may be defined on derived fields, such as the gradient field. Features in these fields are detailed in the respective sections of this thesis.

2.2.1.3. Scalar Field Topology

In the following, we assume that the scalar field is a *Morse Function*, meaning that all critical points are true point-like features and non-degenerate. With this assumption, topological features of a scalar field can be defined as locations where the topology of adjacent level-sets changes for continuous variation of the function value. The resulting topological skeleton is captured by *Morse Theory* [Mil63], which allows decomposition of the scalar field by identification of regions of influence of different extrema. This intuitive description of scalar field topology is obtained by the observation of merging behavior of level-set contours on terrain. If one continuously rises a virtual ”water-level” (function value) in a two-dimensional height-field and tracks individual level-set contours, the three following topology changes can be observed:

- *Creation* of new, disconnected level set contours: The water-level has traversed the function value of a local minimum.
- *Merging* of previously disconnected level set contours: The water-level has crossed the function value of a saddle.
- *Vanishing* of level set contours: The water-level has crossed the function value of a local maximum.

Locations of these events are nodes of a topological graph that segments the field into regions with identical level-set contour limits. A different interpretation of this topological graph by means of the vector-valued gradient field is given in the next section.

Figure 2.6 illustrates the graphical construction of scalar field topology for rising water-levels. This construction segments the data set into regions of influence of the minima of the field. In real-world height-field or terrain analysis, these regions have the meaning of catchment basins, which describe, where water flows downhill to a common location. Analogously, regions of influence are obtained for maxima. The intersection of both decompositions is a complete description of the field topology. A related feature, that tracks topological changes in level-sets is the *Reeb graph* or *Contour tree* of scalar fields [vKvOB$^+$97], which is however not of immediate relevance to the work covered in this thesis.

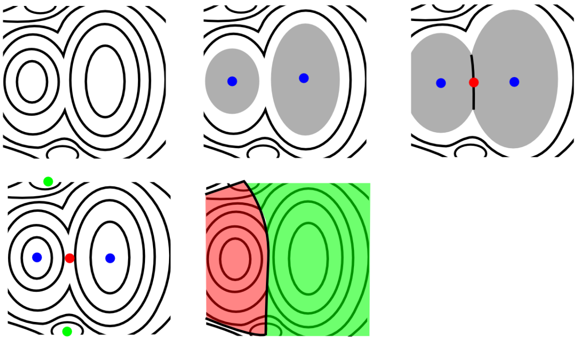


Figure 2.6.: Two-dimensional example of function topology. Events where the level-set topology changes are colored in blue (contour creation event), green (contour vanishes) and red (contours merge). Ascending manifolds of the minima are highlighted as red and green regions.

2.2.1.4. Scalar Field Visualization Techniques

Direct visualization of scalar fields may be performed by the use of transfer functions. In scalar field visualization, an n-dimensional *transfer function* maps values from n scalar fields to a color space such as RGB(A):

$$f : \mathbb{R}^n \rightarrow \mathbb{R}^4_{RGBA}$$

Transfer functions in volume rendering [Lev88, DCH88] can be designed to highlight features such as isocontours or interval sets by assignment of high opacity to respective function values. While application of transfer functions to two-dimensional scalar fields results in flat images, three-dimensional scalar field visualization usually relies on more complex visualization techniques such as volume-slicing or ray-casting to avoid occlusion [Lev90, LL94] in the resulting volume image. Other direct volume visualization techniques include rendering of color-mapped point clouds or particle splatting [CM93].

As some feature definitions such as iso-contours and extremal structures naturally produce surface or line-like scalar field features, it is often more convenient to visualize a tessellated version of this geometry instead of applying a transfer function filter to the whole data set. While early work was focused on surface extraction in structures data sets [LC87], existing algorithms are able to produce triangulations of isosurfaces in fields with arbitrary neighborhoods [HSIW96, AG01]. Advantages of meshed feature visualization include the availability of texturing and shading techniques. However, without additional effort, complex meshes have a tendency to occlude parts of the data set.

2.2.2. Vector Fields

Vector fields extend the notion of scalar fields with directional information. A common interpretation of vector-valued data takes vector magnitude and orientation as speed and direction of motion of infinitesimal particles in space. This view facilitates modeling of natural phenomena such as wind or streams (see Figure 2.7).

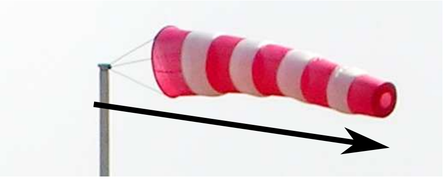


Figure 2.7.: Example of a vector illustration denoting wind direction as abstracted from a real-world measurement.

2.2.2.1. Application Areas

Some measurement techniques such as *Optical Flow* [HS81] estimation from video data for traffic control and tracking or deformation measurements in physics and mechanics [vdGW99] for material analysis and engineering produce vector fields. However, vector field visualization is mainly dominated by the analysis of flow fields resulting from *Computational Fluid Dynamics* (CFD) simulations [Bat67].

Computational fluid dynamics utilize fundamental laws of physics to model the behavior of fluids. Using partial differential equations such as the *Navier-Stokes* equations [Bat67], velocity of a fluid is coupled to (external) forces related to pressure, fluid viscosity, gravity, and stresses acting on fluid elements. Solutions of these Navier Stokes equations correspond to vector-valued flow fields. The increasing need for accurate and flexible flow simulations is fueled by the construction of increasingly complex industrial mixing processes, and virtual prototyping in aviation or the automotive industry. In virtual prototyping, visualization can help in identifying beneficial or undesired aerodynamical properties before testing real prototypes in expensive wind tunnel experiments.

2.2.2.2. Features of Vector Fields

As vector fields are frequently used to model flow, common feature related questions aim at answering the question of *how* matter is flowing in a global or local sense.

A straight-forward attempt to answer this question is to trace the path of a virtual particle as it is advected by the flow field. The resulting curve is a vector field feature and is known as *integral line*. In higher dimensions, this feature can easily be generalized by the advection of curves or surfaces [Hul92].

More specific feature definitions are motivated by their importance to applications.

Since in aviation swirling motion has been identified as one of the key contributors to flight properties such as fuel consumption (drag) or lift, experts from this field primarily interested in this type of spiraling motion of a fluid around a center (-line). Such swirling flow is known as *vortex*.

In industrial mixing, however, relative fluid motion with an impact on mixing quality is of central interest. This motivates the definition of features that characterize regions of diverging or converging flow behavior and analysis of high and low-strain regions in the flow field.

A universally meaningful feature of vector fields are areas where flow vanishes, i.e. the magnitude of the vector field is locally zero. Presence of such *critical points* facilitates extraction of topological graphs (see next section). Furthermore, a number of scalar-valued derived fields (*vorticity*, *deviation*, etc.) contain meaningful features as well.

2.2.2.3. Vector Field Topology

The questions of which flow regions show similar behavior and desired knowledge about flow convergence and divergence leads to the definition of vector field topology. Vector field topology in a strict sense is only defined on stationary vector fields. We therefore limit this section to the description of instantaneous vector field topology and give a mathematical foundation of related concepts in time-varying vector fields in Section 3.5.2.

As discussed in Chapter 3, scalar field and stationary vector field topology share important characteristics. Topological vector field analysis partitions the domain into integral lines that exhibit an identical limit behavior. In the literature, this limit behavior is mathematically described by α (backward) and ω (forward) *limit-sets* of integral lines $s : \mathbb{R} \to \mathbb{R}^n$ (see [WS01]):

$$\begin{aligned}
\alpha(s) &= \{p \in \mathbb{R}^n | \exists (t_n)_{n=0}^{\infty} \subset \mathbb{R}, t_n \to -\infty, lim_{n\to\infty} s(t_n) = p\} \\
\omega(s) &= \{p \in \mathbb{R}^n | \exists (t_n)_{n=0}^{\infty} \subset \mathbb{R}, t_n \to -\infty, lim_{n\to\infty} s(t_n) = p\}
\end{aligned}$$

Locations defined by these limit-sets either take the form of interior critical-points of a vector field, cycles, or the boundary of the data set. To segment the field into regions with homogeneous flow behavior, particle traces that emerge from the same limit-set and lead to the same limit-set are grouped into the same cell of the topological graph. An illustration of these zero-, one-, and two-dimensional manifolds is shown in Figure 2.8. Edges in the topological graph are separating integral flow features known as *separatrices*.

According to this definition, the topological graph of a gradient field coincides with the Morse-Smale complex of its scalar field. However, due to the lack of

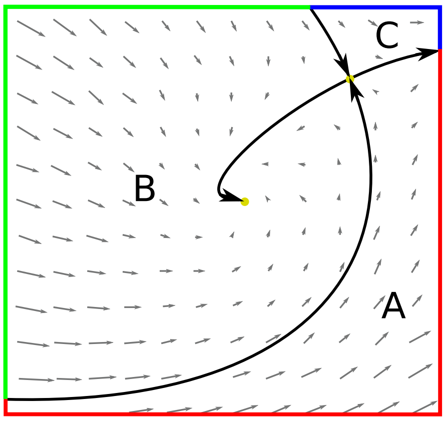


Figure 2.8.: Topological graph of a two-dimensional vector field comprised of separatrices (black) and critical points with a rotating interior sink and an interior saddle (yellow). Topological structures of the field include zero-, one-, and two-dimensional manifolds: Two critical points, four separatrices, and three regions inside and on the boundary of the data set (A,B,C).

rotation in gradient fields, not all topological graphs of vector fields can be interpreted as Morse-Smale complexes of scalar fields.

In time-varying vector fields, snapshots of instantaneous vector field topology allow the definition of stable features and merging or separating behavior of critical points over time also called *bifurcations* [TSH01b]. Practical applications of these stationary views of time-varying vector fields are limited, however. In general, critical points and separatrices of a vector field are referred to its topological features. For small perturbations, the structure of the topological graph remains stable.

2.2.2.4. Vector Field Visualization Techniques

Basic direct vector field visualization methods explicitly draw arrow-shaped vectors at densely sampled positions in the field resulting in a cluttered view of the complete field. Refinements of this method replace arrow-like vector representations by short streamlines or complex glyphs such as flow probes [dLvW93].

Instead of explicitly constructing vector icons at distinct positions in space, related texture based methods such as *Line Integral Convolution* (LIC) [CL93] produce a dense image of flow behavior by blurring a (noisy) input texture along integral flow paths. Sophisticated use of transparency and volume-rendering techniques additionally allows application of texture-based methods to volumetric data sets [IG97].

Especially in time-varying fields, flow-advected Lagrangian particles or textures serve the creation of insightful animations [KKKW05].

With a few exceptions, where methods from illustrative mesh rendering were applied to highlight characteristic features of integral surfaces [HGH$^+$10], visualization of tessellated integral features [GKT$^+$08] usually draws on standard surface rendering and transparency techniques to help convey the three-dimensional structure of the field and minimize (self-) occlusion.

2.2.3. Second Order Tensor Fields

Tensor fields (here limited to second order tensors or *matrices*) in the context of this thesis are used to denote changes or linear mappings between different vectors. As such, they facilitate modeling of changes in shape, direction or position such as deformation in mechanics (see Figure 2.9).

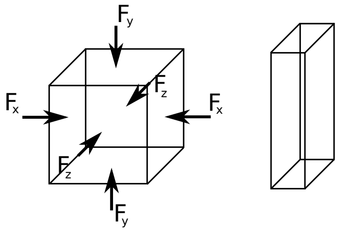


Figure 2.9.: Illustration element deformation caused by application of directional forces. Changes along x,y and z axes may be modeled as deformation tensor.

2.2.3.1. Application Areas

Second order tensor fields are created by simulations and measurements from areas that are concerned with quantities describing diffusion, deformation, displacement or directional anisotropy in general. In visualization, the most frequently visited type of tensor field contains stress and strain tensors, representing the direction and effects of (internal and external) forces on a given continuum.

In medicine, *Diffusion-Tensor MRI* (DT-MRI) [BJ02] is an imaging technique used to measure directional information present in water diffusion. These measurements are usually conducted on parts of the brain, where major diffusion directions are governed by fiber directions. These anisotropy measurements produce symmetric 3 × 3 diffusion tensors on a voxel grid, whose main eigenvector directions correspond to fiber directions. Visualization of this directional information can provide important insights into connectivity of structures in the brain.

Application areas concerned with deformation and displacement are mainly related to material sciences or physics, where significant effort is directed towards modeling, simulating and predicting accurate object deformations. The study of displacement is of great importance in (geo-)physics and mechanics. Derivation of methods to predict forces causing this displacement helps in understanding surface and material behavior and faulting [Koy97].

2.2.3.2. Features of Tensor Fields

The fact that tensors are frequently used to model deformation or other multilinear maps on vectors puts emphasis on the need to analyze analytic properties of this vector transformation.

Eigenvector directions of a matrix have the mathematical property of remaining unchanged under the linear map described by the tensor. The eigenvector with the largest absolute eigenvalue has a dominant meaning in tensor analysis. The result of repeated mapping of an arbitrary vector (that is not an eigenvector) converges to the direction given by this eigenvector [KC00]. Given the importance of these characteristic quantities, it is not surprising that the majority of feature definitions for second order tensor fields relies on eigen-decomposition and related tensor properties such as singularities and degeneracies.

This decomposition of tensor fields into sets of eigenvector fields allows the definition of *hyper-streamlines* and *tensor-lines* [DH92, DH93], which are integral lines of these eigenvector fields. Parts of the literature distinguish between hyper-streamlines and tensor-lines by allowing tensor-lines to follow eigenvector fields smoothly without being forced to a designated eigenvector field, e.g. the major eigenvector direction [WKL99]. This definition of tensor-lines facilitates stable integration through almost isotropic regions.

Based on the concept of these integral lines, fibers are tracked in DT-MRI tractography [BPP$^+$00], where *fiber-bundles* and their crossings are regarded as features. Like in vector field visualization, tensor field operator can produce fields with scalar values such as the tensor determinant etc., whose feature correspond to the feature definitions given in Section 2.2.1.2.

2.2.3.3. Tensor Field Topology

When tensors are modeling material deformation, discontinuities in displacement or deformation behavior are of special interest to domain experts, as they are cause of possible faulting [Koy97].

Tensor field topology analysis is a relatively new area of science when compared to topological analysis of scalar- or vector fields and is mostly limited to symmetric tensors, where eigenvector directions are pairwise orthogonal. A topologically relevant feature in symmetric tensor fields that separates regions of similar eigenvector behavior are regions, where the tensor is (partially) isotropic [DH94]. At these locations, multiple eigenvalues are identical, preventing the unique computation of all eigenvectors as illustrated in Figure 2.10.

Again, the concept of a topological skeleton along with separating manifolds is transferred from scalar- and vector field topology and allows definition of separa-

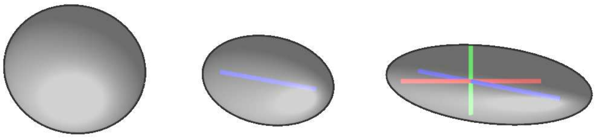


Figure 2.10.: Ellipsoids can be used to illustrate deformation mappings of symmetric tensors. Isotropic tensors do not have uniquely determined eigenvector directions (left). Partially degenerate tensors allow computation of one unique eigenvector direction (middle). Fully anisotropic tensor has three unique eigenvector directions (right).

tion structures in tensor fields by construction of integral surfaces emanating from these *degenerate lines* and *degenerate points* by advection of these features along eigenvector directions. The resulting regions indicate homogeneous behavior of the dominant eigenvector direction.

2.2.3.4. Tensor Field Visualization Techniques

Direct tensor field visualization methods rely on modeling single tensors as *tensor glyphs*. For symmetric tensors, the most basic glyph [PvWPS95] is an ellipsoid whose main axes correspond to tensor eigenvectors scaled by eigenvalue magnitude. Other tensor glyphs facilitate visualization of asymmetric tensors and put emphasis on different tensor properties to characterize and distinguish anisotropy classes and eigenvector directions [Hab90, MSM96, Kin04]. Feature based tensor field visualization is traditionally focused on vector field type visualizations of the derived eigenvector fields, which results in the display of lines of principal curvature/eigenvector directions with standard or enhanced vector field visualization techniques (e.g.: [DH93, DH94, HFH$^+$04]). Topology-based visualization techniques create graph-like structures, whose geometry may be visualized directly, or after simplification [TSH01a].

2.2.4. Multi-Field

Multi-field data contains multiple (related) fields of identical or different types and dimensionality.

2.2.4.1. Application Areas

Multi-field visualization is relevant to virtually all application areas, as applications either directly produce multiple fields or one can derive multiple related fields from the given single field data.

Computational fluid dynamics simulations are an example of the former type, as they naturally generate a set of related fields (e.g: pressure, temperature, velocity) [Bat67]. In applications, where only one field is generated, analysis of derived fields often helps in understanding problems at hand and identify important relationships.

2.2.4.2. Multi-Field Features

The main insight that one hopes to gain by the extraction of features in multi-field data is the identification of field interactions and relationships. Features in multi-field data are either *feature combinations* or true *multi-field features*.

The first type of feature definition is obtained by combining existing feature definitions of different field types. These feature combinations include common boolean operators on existing feature definitions such as AND, OR and XOR [KPI+03]. An example of such a feature is obtained by asking, where maxima of multiple scalar fields coincide. Such feature definitions have in common that the initial feature definition and extraction process is limited to a single field and is followed by a post-processing step to obtain an instance of the combined feature. Therefore, this type of feature definition is covered by previous sections on scalar-, vector-, and tensor field features.

True multi-field features require data from all affected fields during the feature extraction process. The parallel-vectors operator [PR99], is an example of such a feature defining features as regions, where multiple-vector fields are parallel.

2.2.4.3. Multi-Field Visualization

Multi-Field visualization makes use of the visualization techniques presented in previous sections. In general, multiple fields can be visualized by combining direct visualization techniques (such as icons, glyphs and volume rendering) [KPI+03], by a combination of feature-visualization techniques [UI08], or mixtures of these [UIL+06].

With the increase of data and feature density in data sets that contain multiple fields, special care has to be taken to avoid the occlusion of important features of one or more fields. This thesis shows two ways of reducing visual occlusion, namely region extraction/segmentation of a primary field and combination with direct field visualization and the use of multiple linked views.

2.3. Field Representation

The digital nature of computers requires that a (piecewise) continuous n-dimensional tensor-valued signal or field $f : \Omega \subseteq \mathbb{R}^n \rightarrow \mathbb{R}^{m_1 \times \cdots \times m_j}$ is sampled at discrete positions $x_i \in \Omega$ during simulation or measurement. The resulting set of points (p_i, f_i) with $f_i = f(p_i) \in \mathbb{R}^{m_1 \times \cdots \times m_j}$ represents the data available for post-processing and visualization. While there are ways to record and store analog data, such data sets nowadays represent the exception to the rule in computer science.

In most cases this point set is accompanied with concrete neighborhood information either for post-processing convenience or because the simulation techniques used for creation of the data themselves rely on grids. These neighborhood graphs or grids may be of one of the following two general types:

- structured (implicit, cartesian, curvilinear, ...)
- unstructured (tetrahedral, hexahedral, mixed elements, ...)

Scattered point sets on the other hand do not possess such an explicit *computational mesh.* Structured grids are composed of a single element type, e.g. cubes, and their regular layout allows implicit storing of the neighborhood relation. Unstructured grids may be composed of a number of different element types and often consist of irregularly sized and oriented cells, which are necessary to represent complex geometry or adaptive point set densities, and require explicit definition of the neighborhood data.

Reconstruction of a (continuous) field from the given point set at arbitrary positions $p \in \Omega$ requires definition and evaluation of the neighborhood of p. Consequently, the nature of the underlying grid structure has a central impact on the type, performance, and mathematical properties of the available field reconstruction methods.

2.4. Field Reconstruction

Given a discrete (discontinuous) field representation as point-set (p_i, f_i), the goal of field reconstruction methods is to (re-)construct function values of the field that are not provided by the given samples. This reconstructed function f may either *interpolate* $(f(p_i) = f_i \quad \forall i)$ or *approximate* $(\exists i : \quad f(p_i) \neq f_i)$ the set of given function values. Commonly, the reconstruction function is expected to have certain properties such as C^m continuity. It is important to note that interpolation and approximation techniques generally do not generate ”new” information about a field, as choice of interpolation method and parameters is non-unique in all cases, where the interpolant of the original data is unknown.

In a grid-based field, function evaluation at point p amounts to locating the cell that contains p along with choosing an appropriate element-based interpolation or approximation method such as trilinear interpolation for cubes, barycentric interpolation for tetrahedra, or higher-order grid-based techniques. As most parts of this thesis either operate on scattered data or are independent of the underlying interpolation method, we do not go into further detail about grid-based interpolation methods, but refer to the literature [LGS99].

Scattered data may be approximated by grid-based methods after construction of a tessellation. Without such a tessellation, the absence of a pre-defined neighborhood relation prevents the use of element-based approximation techniques and requires the definition of a neighborhood for the data set [Ami02]. The central advantage of grid-less approximation is the independence of a computational grid, i.e. that solutions are governed by the field's value rather than by the choice and characteristics of a (static) neighborhood structure. Moreover, concrete neighborhood structures for a field may not be defined in a unique way (cf.: decomposition of cubes into tetrahedra) and may have to change over time to adapt to transformations of the domain. However, as already mentioned, faithful reconstruction of a data set is only possible if the same interpolation method is used during creation, i.e. simulation of the phenomenon, and visualization of the data set. Other scattered data approximation techniques are, for example, discussed in [Wen04]. A large group of scattered data approximation methods falls into the group of *Radial Basis Function* (RBF) approximation techniques [Wen04], which use (euclidean) distance for neighborhood weighting purposes. The general form of a RBF interpolant for a point set (p_i, f_i) is the linear combination of radial functions

$$f(p) = \sum_i \omega(\|p - p_i\|)v_i$$

where $\omega : \mathbb{R} \to \mathbb{R}$ is a radial function defining the degree of neighborhood between point p and p_i. The weights v_i are determined by the chosen interpolation or approximation conditions. In these scattered data approximation techniques, neighborhood is defined by a distance metric and is otherwise independent of other spatial properties of the data points.

Moving Least Squares as the main approximation technique used in this thesis is closely related to RBF approximation. Appendix A gives a definition of MLS and provides an overview of its mathematical and numerical properties.

3. Mathematical Definitions, Properties, and Methods

The work presented in this thesis makes frequent use of several mathematical definitions and properties from fields such as *Linear Algebra* and *Differential Geometry*. To set up a well-defined mathematical framework, the following sections introduce the relevant mathematical fundamentals that are used throughout this thesis and provide details of numerical computation methods, as found in the literature [Ros00, LRK10, Gin09].

3.1. Notation and Prerequisites

If not explicitly stated otherwise, all definitions and properties given in the following assume n-dimensional euclidean space with scalars in $\mathbb{R}$. As defined by Einstein Notation, multiple occurrences of indices in a term of an equation denote summation over the range of the index. If not defined otherwise, x, y, z and x_i, $i \in \mathbb{N}^+$ denote cartesian coordinates with points $p = (x_1 \dots x_n)^T \in \mathbb{R}^n$. Norms commonly denote the standard L^2 norm.

3.2. Linear Algebra and Vector Calculus

Vector-spaces and linear mappings between them are studied in the field of Linear Algebra. Differentiation and integration of vector fields is covered by the branch of mathematics known as Vector Calculus. Both areas of mathematics play a major role in all parts of this thesis and facilitate formulation of the most fundamental definitions of tensors and fields.

Definition 3.1 (Tensors, Scalars, Vectors) *A* tensor *of order m in $\mathbb{R}^{n_1 \times \dots \times n_m}$ has $\prod_{j=1}^{m} n_j$ components $t_{i_1 \dots i_m} \in \mathbb{R}$, with indices $i_j \in \{1, \dots, n_j\}$. Tensors of order* 0*,* 1*, and* 2 *are known as* scalars, vectors*, and* matrices.

A tensor can be written as a n-dimensional array, whose size is indicated by $n_1 \times \cdots \times n_m$ or $T_{n_1 \times \dots \times n_m}$ in the following. In a more general definition of tensors, individual components are allowed to be complex numbers in $\mathbb{C}$. All tensors covered in the context of this thesis are of order ≤ 2. This constraint includes scalar, vector and typical stress or strain tensor fields. For an overview of the most basic tensor characteristics such as *rank* and *determinant*, we refer to standard literature on linear algebra [Ros00] and focus on operators and properties of immediate relevance to field visualization in the following.

3.2.1. Fields, Operators

Definition 3.2 (Tensor Field) *A time-varying* tensor field *is represented as a tensor-valued function*

$$\begin{aligned} f : I \subseteq \mathbb{R} \times \Omega &\rightarrow \mathbb{R}^{n_1 \times \ldots \times n_m} \\ (t, p) &\longmapsto f(t, p) \end{aligned}$$

defined in $(n+1)$*-dimensional space of time* $t \in I \subseteq \mathbb{R}$ *and euclidean space* $\Omega \subseteq \mathbb{R}^n$*. For fixed t,* f *is a* stationary *or* steady *field.*

Respective definitions are given for scalar- and vector fields.

Definition 3.3 (Isotropic and Deviatoric Components) *A symmetric second-order tensor* $T_{n \times n}$ *may be decomposed into an* isotropic *and a* deviatoric *part*

$$T = T_{iso} + T_{dev} = \frac{1}{n} \sum_i t_{ii} \cdot I + \left(T - \frac{1}{n} \sum_i t_{ii} \cdot I \right).$$

Tensors with a deviatoric component of $0_{n \times n}$ are *isotropic*, otherwise they are *anisotropic*. Mathematical operators from vector-calculus allow mapping between tensor fields of different order and are important in the context of visualization of multiple fields. A selection of important operators is presented in the following. Best known function operators from the field of calculus are related to differentiation.

Definition 3.4 (Differential Operator ∇, Gradient) *The differential operator in* $\mathbb{R}^n$ *denoted by* ∇

$$\nabla = \sum_{j=1}^{n} \frac{\partial}{\partial x_j} e_j$$

for canonical cartesian basis vectors e_j *is used to define the* gradient

$$\nabla f = \left(\frac{\partial f}{\partial x_1}, \ldots, \frac{\partial f}{\partial x_n} \right)$$

of a scalar-valued differentiable function f *defined on* $\mathbb{R}^n$*. For a scalar field* f*,* ∇f *denotes the vector-valued* gradient field*.*

The gradient of a scalar field at p is oriented along the direction of maximal ascent of the function and is perpendicular to the tangent of the level set $\{p' \in \mathbb{R}^n | f(p') = f(p)\}$. A gradient magnitude of zero is an indicator of an extremum or a plateau in a scalar field. In general, differentiation of multi-variate tensor fields of order n leads to a tensor field of order $n+1$. Thus, first order derivatives can provide a notion of component-wise change in arbitrarily high order fields. A generalization of the gradient definition is known as the *Jacobian.*

Definition 3.5 (Jacobian) *The* Jacobian *of a differentiable vector field*

$$\begin{aligned} v : \Omega \subseteq \mathbb{R}^n &\rightarrow \mathbb{R}^m \\ p &\longmapsto (v_1(p), \dots, v_m(p)) \end{aligned}$$

is defined as

$$J = \begin{pmatrix} \nabla v_1 \\ \vdots \\ \nabla v_m \end{pmatrix} = \begin{pmatrix} \frac{\partial v_1}{\partial x_1} & \cdots & \frac{\partial v_1}{\partial x_n} \\ \vdots & & \vdots \\ \frac{\partial v_m}{\partial x_1} & \cdots & \frac{\partial v_m}{\partial x_n} \end{pmatrix}$$

and takes the form of a second order tensor field.

The Jacobian is is in general not a symmetric or square matrix. As the Jacobian corresponds to component-wise derivation, its rows are gradient vectors of the scalar fields $v_1, \dots, v_m$.

Definition 3.6 (Hessian) *The* Hessian *of a twice-differentiable scalar field f is a square matrix containing second order derivatives*

$$H = \begin{pmatrix} \frac{\partial^2 f}{\partial x_1^2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \vdots & & \vdots \\ \frac{\partial^2 f}{\partial x_1 \partial x_n} & \cdots & \frac{\partial^2 f}{\partial x_n^2} \end{pmatrix}$$

and is symmetric for f with continuous derivatives.

The Hessian of a scalar field corresponds to the Jacobian of its gradient field. If the second derivatives are continuous, the Hessian matrix is a symmetric matrix. Like any other second order tensor, the Jacobian and the Hessian define linear mappings on vector spaces. Properties of these mappings are of great interest for vector and tensor field analysis and visualization. Several mathematical quantities allow characterization of these mappings.

Definition 3.7 (Eigenvalues, Eigenvectors) Eigenvectors e_i *of a second order square tensor $T_{n \times n}$ are non-zero n-dimensional vectors, whose transformation under the linear mapping defined by T degenerates to scaling:*

$$T \cdot e_i = \lambda_i \cdot e_i.$$

Corresponding scalars λ_i are called eigenvalues *of T.*

A $n \times n$ matrix has at most n distinct eigenvalues which are solutions to the characteristic equation $det(T - \lambda \cdot I) = 0$. A matrix is *singular* if one of its eigenvalues is 0 ($det(T) = \prod_i \lambda_i = 0$). In the case of multiple identical eigenvalues (tensor is (partially) isotropic), directions of the corresponding eigenvectors are non-unique. Eigenvectors for distinct eigenvalues are linearly independent and invariant with respect to scaling, including change of orientation. If all of its eigenvectors are linearly independent, a matrix is *diagonalizable*. For symmetric second order tensors they are orthogonal and eigenvectors as well as eigenvalues are real-valued. The concept of eigenvectors allows classification of tensors according to their definiteness.

Definition 3.8 (Definiteness) *A second order square tensor $T \in \mathbb{R}^{n \times n}$ is* definite*, if for all non-zero $x \in \mathbb{R}^n$*

$$\begin{aligned} x^T T x > 0 &\quad \text{(positive definite)} \\ x^T T x < 0 &\quad \text{(negative definite).} \end{aligned}$$

T is semidefinite*, if for all non-zero $x \in \mathbb{R}^n$*

$$\begin{aligned} x^T T x \geq 0 &\quad \text{(positive semidefinite)} \\ x^T T x \leq 0 &\quad \text{(negative semidefinite).} \end{aligned}$$

If none of these cases apply, T is indefinite*.*

For symmetric matrices an equivalent definition of definiteness is given by the signs of the eigenvalues. An operation related to eigenvalue decomposition that can be applied to non-square tensors is given by the *Singular Value Decomposition* (SVD) [MMH04] .

Definition 3.9 (Singular Values) Singular values *of a second order tensor $T_{n \times n}$ are the square roots of eigenvalues of $T \cdot T^H$, where T^H denotes the conjugate transpose of T.*

Eigenvectors of $T \cdot T^H$ define an orthonormal basis for $T(.)$. For symmetric real matrices, these vectors are identical to the eigenvectors of T. Geometrically speaking they correspond to the half-axes of an ellipsoid created by deformation of a sphere by the mapping T. This interpretation suggests the use of singular values to measure the effect of deformation. One such measure of deformation is given by the Frobenius norm.

Definition 3.10 (Frobenius Norm) *The* Frobenius Norm *of a second order tensor T denoted as $||.||$ corresponds to the square root of the trace of $T \cdot T^H$*

$$||T|| = \sqrt{tr(T \cdot T^H)}$$

In the following, *tensor* and *tensor field* refers to the space of second order quadratic tensors in $\mathbb{R}^{n \times n}$.

3.2.2. Numerical Methods

3.2.2.1. Numerical Differentiation

While interpolation or approximation methods may yield analytically differentiable or integrable representations of fields, higher order analytic derivatives are commonly not available, especially when low-order polynomials are used for reconstruction. Numerical differentiation methods allow approximation of these derivatives.

Definition 3.11 (Difference Quotient) *The first order* forward difference quotient *of a function f at t is defined as*

$$\frac{f(t+h)-f(t)}{h}.$$

The derivative of a function is the limit of the first order difference quotient for $h \to 0$. Therefore, this difference quotient approximates f' for small h. Higher order derivatives may be approximated by according higher order difference quotients, leading to increased number of required function evaluation.

3.2.2.2. Tensor Decompositions

If the roots of the characteristic polynomial $p(\lambda) = det(T - \lambda I)$ are difficult to find analytically, i.e. $p = 0$ is complex to solve for λ ($n > 3$), one relies on numerical methods to compute eigenvalues and -vectors. One such popular method is the *QR-decomposition* with complexity in $O(n^3)$.

As pointed out, the square roots of the eigenvalues of a $T \cdot T^H$ are the singular values of a matrix T. The eigenvectors of $T \cdot T^H$ are the corresponding left singular vectors, the eigenvectors of $T^H \cdot T$ the right singular vectors. This relationship allows numerical computation of singular values and vectors with eigen-decomposition methods. Alternatively, the Singular Value Decomposition provides a tensor decomposition that yields singular values and (left/right) singular vectors directly.

3.3. Continuum Mechanics

The field of *Continuum Mechanics* [LRK10] combines the mechanics of fluids and solids on a macroscopic level, i.e. sizes of atomic elements are small with respect to the size of the continuum. In this thesis, we focus on fluid motion as well as kinematic forces leading to deformations and displacement. The following sections concentrate on the physical interpretation of vector fields as velocity or displacement fields and introduce necessary mathematical concepts and definitions.

3.3.1. Fluid Mechanics

Continuum mechanics allow the modeling of complex fluid behavior in the form of (continuous) fields. Thus, research in *Fluid Mechanics* is concerned with the mathematical modeling and analysis of forces in fluids, complex interactions, and resulting quantities such as viscosity, density, and velocity. As a consequence, simulation data from fluid dynamics commonly provides multiple tensor fields of different order.

Definition 3.12 (Velocity Field, Flow Field) *A vector field representing the motion of a fluid is known as* velocity field *or* flow field.

An exception to this definition is given by *granular flow*, which may by modeled by vector fields as well. Generally, any differentiable vector-valued function can be interpreted as describing particle motion. Fluid dynamics know two ways to describe flow: The *Eulerian* and the *Lagrangian* view. While the Eulerian specification defines fluid flow as a time-varying vector field given at fixed positions in space, the Lagrangian view specifies flow as the motion of particles. These different concepts are reflected in fluid simulation meshes, where explicit grids are commonly used for static Eulerian specification of flow and scattered point sets of moving points are chosen to represent Lagrangian motion, allowing the efficient modeling of free surfaces without the need of frequent remeshing of the domain (see Figure 3.1).

Definition 3.13 (Particle Motion) *A velocity field $v : \mathbb{R} \times \mathbb{R}^n \rightarrow \mathbb{R}^n$ describing the motion of a fluid particle with position $x(t) \in \mathbb{R}^n$, $t \in \mathbb{R}$ is governed by the ordinary differential equation (ODE)*

$$\frac{dx(t)}{dt} = v(t, x(t)). \tag{3.1}$$

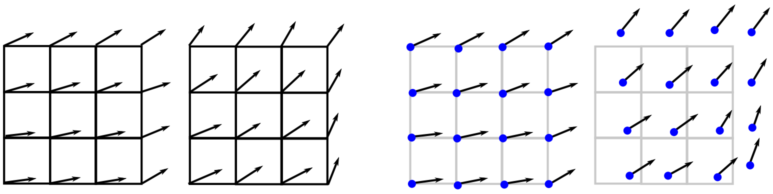


Figure 3.1.: Two different specifications of the same flow field at two subsequent time steps. Left: Eulerian specification with velocity specified at fixed positions in space. Right: Flow field specified at Lagrangian particles (corresponding Eulerian grid shown in gray).

If t is constant/parametrizes a spatial dimension in $\mathbb{R}^n$, the velocity field $v(x) := v(t, x)$ is time-independent and represents *steady flow*. Otherwise it is a *time-*

varying vector field. Thus, a time-varying vector field in $\mathbb{R}^n$ can be interpreted as stationary field in $\mathbb{R}^{n+1}$ [TS03].

For flow fields, scalar measures are defined that facilitate computation of local flow behavior. The most notable scalar measures are able to quantify volume change or rotation.

Definition 3.14 (Curl) *The vector-valued rotation of a vector field v is known as* curl

$$curl(v) = \nabla \times v.$$

Fields with $\nabla \times v = 0$ *are* irrotational.

In two-dimensional vector fields, *curl* is a scalar quantity. Gradient fields are by definition irrotational, as $\nabla \times (\nabla v) = 0$. This constitutes a central difference between general vector fields and gradient fields.

Definition 3.15 (Divergence) *The trace of the Jacobian of a vector field v is known as* divergence

$$div(v) = \nabla \cdot v = \sum_i \frac{\partial v_i}{\partial x_i}.$$

Fields with $\nabla \cdot v = 0$ *are* divergence-free *or* incompressible.

Divergence serves as a measure of volume change in vector fields. As a consequence, no fluid matter is created or destroyed in divergence-free flow fields. Volume change and distortion are concepts closely related to strain and displacement. The next section gives further details about the relationship between displacement and flow fields.

3.3.2. Solid Mechanics

While forces acting upon fluids result in fluid motion as defined by the Navier-Stokes equations, effects of forces on continuum bodies include deformation and displacement. While some definitions of deformation are extended to rigid bodies, we limit this notion in the following to bodies where deformation causes relative geometric displacement [vdGW99], i.e. a change in shape. This section aims to give an introduction to (flow induced) strain.

3.3.2.1. Deformation

Definition 3.16 (Displacement Function) *In continuum mechanics, a* displacement function *u maps each point $p \in \Omega \subseteq \mathbb{R}^n$ within a continuum body Ω to a position $p' \in \mathbb{R}^n$.*

$$u : \Omega \subseteq \mathbb{R}^n \rightarrow \mathbb{R}^n$$

Displacement functions are vector fields. A common requirement of displacement functions is that domain and range are n-dimensional vector-spaces. The Jacobian of u is known as *displacement gradient.*

Definition 3.17 (Strain) Strain *is a measure denoting the relative geometric displacement of positions in an object or medium.*

In contrast to absolute geometric displacement, which leads to object translation or rotation, relative geometric displacement within a body causes geometric deformation. This definition allows distinction of rigid and non-rigid behavior of bodies.

Strain as measure of geometric displacement is directly related to first order derivatives of the displacement function, as mentioned for volume change in flow fields in the previous section.

Definition 3.18 (Lagrangian Strain Tensor) *The* Lagrangian Strain Tensor *for a differentiable displacement function u is defined as*

$$e_{ij} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} + \frac{\partial u_k}{\partial x_i}\frac{\partial u_k}{\partial x_j}\right).$$

For small deformations $\frac{\partial u_k}{\partial x_i}\frac{\partial u_k}{\partial x_j}$ vanishes and the Lagrangian Strain Tensor can be approximated by a linearization.

Definition 3.19 (Infinitesimal Strain Tensor) *For a differentiable displacement function u with $\|u\| \ll 1$, $\|\nabla u\| \ll 1$, the components of the second order Infinitesimal* Strain Tensor *are defined as*

$$\epsilon_{ij} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right).$$

Definition 3.20 (Infinitesimal Rotation Tensor) *Similar to the infinitesimal strain tensor, the Infinitesimal* Rotation Tensor *is defined as*

$$r_{ij} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} - \frac{\partial u_j}{\partial x_i}\right).$$

This tensor describes a rotation around $\nabla \times u$ and is therefore related to the curl of vector fields.

This infinitesimal deformation tensor is a linearization of Lagrangian/Eulerian strain. With these definitions, flow fields can locally be interpreted as a combination of absolute and relative geometric displacement, leading to a change in position and orientation as well as deformation of participating media. More precisely, the flow Jacobian contains information of relative displacement and may be decomposed into the Infinitesimal Strain Tensor and the Infinitesimal Rotation Tensor:

$$(\nabla u)_{ij} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right) + \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} - \frac{\partial u_j}{\partial x_i}\right) \tag{3.2}$$

Types of strain are traditionally classified by their effect into

- *tensile strain*: positive displacement along an object normal
- *compressive strain*: negative displacement along an object normal
- *shear strain*: displacement orthogonal to object normals

These definitions of strain are repeatedly used in the context of geophysical strain analysis as defined in the following section.

3.3.3. Moments

In a physical setting, shear-type vector displacement in inelastic deformations may cause surface faulting. Geoscience uses a mathematically well-founded way of capturing these displacement discontinuities in the form of force couples in moment tensors [Koy97]. While in classical mechanics, a moment is usually a scalar or vector-valued quantity denoting a force and an arm length, the notion of distinct force couples leads to the definition of a moment tensor.

Definition 3.21 (Force Couple) *A* force couple *is a pair of parallel but opposing forces with equal magnitude.*

Generally, a force couple results in rotational motion around an axis when the forces are applied to different positions. A convenient form of notation for a set of orthogonal force couples obtained by seismic measurements is the moment tensor.

Definition 3.22 (Moment Tensor) *The* moment tensor *M is a symmetric second order tensor, whose components m_{ij} denote the magnitude of the moment caused by the force couple in i-direction acting on the the j-direction.*

The moment tensor is symmetric by construction, as the total angular moment of a seismic source is zero ($m_{ij} = m_{ji}$).

3.3.4. Numerical Methods

Fluid motion, strain, and deformation of a continuum are in practice computed on a spatial discretization of the domain. Methods can either use explicit neighborhood grids and are called *Finite Element Methods* (FEM) or *Finite Volume Methods* (FVM) [ZTTZ05], or may be mesh-free. Latter methods are termed *Smoothed Particle Hydrodynamics* (SPH) or *Finite Pointset Methods* (FPM) [TK02] for fluid dynamics and are of central interest in this thesis. In all techniques, PDEs or ODEs are specified at discrete point positions to approximate the full system equation.

In the case of strain or deformation computation, expensive discretization and solution of complex systems can be avoided if the deformable object can be approximated as infinitesimal point.

3.4. Differential Geometry

Differential Geometry is a branch of geometry that defines configuration, shape, and other properties of space by means of differential and integral calculus. For vector field analysis, differential geometry provides means to define integral flow features as discussed in the following.

3.4.1. Integral Features

Differential geometry allows the definition of geometric flow features that correspond to solutions of initial value problems for the underlying ordinary differential equations. For a class of ODEs these solutions are unique.

Definition 3.23 (Initial Value Problem, Solution) *A solution to an* initial value problem *(IVP) is a function x that satisfies an ordinary differential equation*

$$\frac{dx}{dt} = v(t, x(t))$$

with the initial condition $x(t_0) = x_0$.

Definition 3.24 (Lipschitz Continuity) *A vector-valued function $v : I \subseteq \mathbb{R} \times \Omega \to \mathbb{R}^n$ is* Lipschitz Continuous *in $\Omega \subseteq \mathbb{R}^n$ if there exists a non-negative $c \in \mathbb{R}$ for arbitrary $t \in \mathbb{R}$ and $p_1, p_2 \in \Omega$ such that*

$$\|v(t, p_1) - v(t, p_2)\| \leq c \cdot \|p_1 - p_2\|.$$

A function v is locally *Lipschitz if it is Lipschitz Continuous in a neighborhood of p for all $p \in \mathbb{R}^n$.*

Lipschitz continuity guarantees that solutions to initial value problems of the ODE exist and are unique. The weaker criterion of local Lipschitz continuity guarantees that a local solution can be continued to the boundary of $I \times \Omega$. As practical vector fields are defined on closed domains, this ensures existence and uniqueness of solutions. These properties facilitate the definition of unique integral flow features as solutions of ODEs.

Definition 3.25 (Integral Curve) *The* integral curve

$$x(t) = x_0 + \int_{t_0}^{t} v(\tau, x(\tau))\, d\tau$$

solving 3.1 with $x_0 = x(t_0) \in \mathbb{R}^n$ *is a* pathline *in time-varying velocity fields* v *or a* streamline *in steady flow.*

Integral curves trace the path of an infinitesimal particle in velocity fields. Streamlines are by definition tangential to the (stationary) flow field and with the exception of cycles do not self-intersect. Pathlines are no longer tangential to the flow field and may self-intersect. These definitions can be generalized to higher dimensions.

Definition 3.26 (Integral Surface, Rake) *The* integral surface

$$S(s,t) = c(s) + \int_{t_0}^{t} v(\tau, S(s,\tau)) d\tau$$

with a univariate seeding curve or rake $c : I \subseteq \mathbb{R} \to \mathbb{R}^n$ *is a* path surface *in time-varying velocity fields* v *or a* stream surface *in steady flow.*

If c varies over time, this definition characterizes a *generalized path surface.* Path surfaces are parametrized along the seeding curve and the time axis. Lines on a path surface for fixed t are *timelines.* Higher-dimensional seeding geometry generalizes these definitions to *path volumes* and *time surfaces.*

Another definition of integral lines and surfaces in time-varying flow fields is given by streaklines and streak surfaces. A *streakline* is the locus of a set of particles that are advected by a time-dependent flow field and emerge from a pre-defined seeding location or rake.

Definition 3.27 (Streak Surface) *The integral surface* S *defined by*

$$S(r,s,t) = c(s) + \int_{t-r}^{t} f(S(r-(t-\tau), s, \tau), \tau) d\tau$$

is a streak surface *at time* t *with particles emerging from points* $c(s)$ *at an univariate seeding curve* $c : [0,1] \to \mathbb{R}^3$*. Individual instances of particles are identified by their age parameter* $r \in [0,t]$*. For constant* s*,* S *is a* streakline.

Similarly, streak surfaces can be used to model flow regions that pass through a flow region by using backwards integration.

3.4.2. Numerical Methods

Analytic integration for the computation of flow trajectories is usually not possible for the given data. During numerical integration, the interval of an integral is discretized into segments.

Definition 3.28 (Riemann Sum) *The* Riemann Sum *of a given integral* $I = \int_a^b f(x)dx$ *corresponds to a discretization of the interval* $[a, b]$ *into* n *subintervals* $[x_i, x_{i+1}]$ *with* $x_1 = a, x_n = b, t_i \in [x_i, x_{i+1}]$

$$I^* = \sum_{i=1}^{n} h_i \cdot f(t_i)$$

where $h_i = x_{i+1} - x_i$.

As the lengths of the sub-intervals approach 0, the Riemann Sum converges to the integral. From a computational point-of-view, however, evaluation of the Riemann Sum becomes unfeasibly expensive for $n \to \infty$ and approximation quality undesirably inaccurate for large h_i and non-linear f. Fortunately, there are a number of numerical integration methods with provable error boundaries for different numbers of required function evaluations and approximation accuracy. A common property of these integration methods presented in the following is their approximation of the Taylor Series Expansion with low order derivatives.

Definition 3.29 (Taylor Series) *The* Taylor Series Expansion *at* x *of a (locally) infinitely differentiable function* f *in the neighborhood of* p *is*

$$T_p(x) = \sum_{i=0}^{\infty} \frac{f^{(i)}(p)}{i!}(x-p)^i$$

where $f^{(i)}$ *is the* i*-th derivative of* f.

The Taylor Series defines function values in a neighborhood of an evaluation point by function derivatives and can therefore be directly connected to initial value problems. As high order derivatives are commonly not available for discrete vector fields, approximation methods for this series try to replace higher order derivatives by combinations of low order function derivatives.

Definition 3.30 (Euler Method) *A first order numerical integration scheme for univariate* $x : \mathbb{R} \to \mathbb{R}^n$ *is given by the* Euler Method

$$x(t + h) = x(t) + h \cdot x'(t).$$

The Euler Method is a linearization of the Taylor series expansion. The truncation error (the remainder of the Taylor Expansion) per step lies therefore in $O(h^2)$. With $1/h$ steps for one unit of the derivative, the error of the Euler method is $O(h)$ and the method is therefore a first order method. Higher order approximation schemes, such as fourth-order Runge-Kutta methods reduce approximation order by approximation of higher function derivatives in the Taylor series. Methods of numerical integration allow efficient iterative approximation of integrals and therefore solution of IVP while constraining the mathematical order of approximation errors. For an inappropriately large choice of h, however, the error of high-order numerical integration schemes is still undesirably large. A solution to this problem is the automatic selection of appropriate step-sizes h during integration.

In general, adaptive integration schemes such as step-doubling or Runge-Kutta 4/5 [CK90] compute $x(t+h)$ for different h or with varying approximation order to estimate the current influence of h on the accuracy of trajectory reconstruction. As long as the error (e.g.: angular deviation, see (3.3) between different evaluations is too big, computations are repeated with a decreased stepsize h. While leading to an increased number of field evaluations, accuracy of such adaptive trajectories is higher in turbulent regions.

$$E(t; h_0, h_1) = 1 - (x(t + h_0) - x(t)) \cdot (x(t + h_1) - x(t)) \tag{3.3}$$

3.5. Field Topology

As introduced in Chapter 2 from an application point-of-view, field topology is suitable to perform segmentation based on homogeneous field behavior. This section concretizes previous high-level descriptions by providing a formal mathematical presentation of field topology.

3.5.1. Scalar Field Topology

Topology of a scalar field is strongly connected to the location of extremal function values. Definition of higher-dimensional stationary points is directly related to concepts from one-dimensional function analysis.

Definition 3.31 (Critical Points) *A* critical point x *of a differentiable scalar field* f *is defined by*

$$\nabla f(x) = 0.$$

Critical points x of a scalar field may be classified according to the signs of principal curvature of f at x.

Definition 3.32 (Minimum, Maximum, Saddle) *If the Hessian of a scalar field is non-singular, a critical point is a (i)* maximum*, (ii)* minimum *or (iii)* saddle*, if eigenvalue signs of the Hessian are (i) all negative, (ii) all positive, (iii) negative and positive.*

Critical points with singular Hessian mark flat spots and represent line-like degenerate critical-points or (partial) plateaus. Figure 3.2 illustrates different types of critical points.

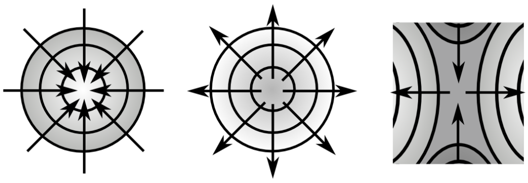


Figure 3.2.: Three critical point configurations in scalar fields: Maximum, Minimum, Saddle. Arrows indicate gradient direction.

Definition 3.33 (Ascending and Descending Manifolds) *The set of all positions that lie on gradient lines starting in a critical point p are the* ascending manifold *of p. The set of all positions that lie on gradient lines ending in a critical point p are the* descending manifold *of p.*

In two-dimensional scalar fields ascending and descending manifolds may be zero-dimensional (the ascending manifold of a maximum is the position of the maximum itself), one-dimensional (the ascending manifolds of a saddle are separatrices), or two-dimensional (ascending manifold of a source). In the context of scalar fields, slope lines corresponding to manifolds of the saddles are often regarded as ridges or valleys in the scalar height-field, as detailed in Chapter 4.

Definition 3.34 (Morse-Smale Complex) *The* Morse-Smale *complex is the intersection of all ascending and descending manifolds of a scalar field.*

The Morse-Smale complex [Zom05] represents the topological graph of scalar fields, whose edges are ascending and descending manifolds of saddle points. Occasionally, discrete Morse Theory in combinatorics is known as the *Theory of Forman.*

3.5.2. Vector Field Topology

Since topology analysis on scalar fields mainly operates on the scalar field's gradient field, stationary vector field topology is closely related to scalar field topology. However, there are notable differences between gradient field topology and general vector field topology, which are detailed in the following sections.

While attempts have been made to define topological graphs for time-varying vector fields, vector field topology is so far only well-defined for stationary fields. For this reason, the following sections distinguish between stationary vector field topology and the related concept of hyperbolicity [Hal01] in time-varying vector fields.

3.5.2.1. Stationary Vector Fields

In contrast to gradient fields, general vector fields may contain rotational components. With this property in mind, the critical point definitions in scalar fields carry over to vector fields.

Definition 3.35 (Critical Points) *A* critical point *or* stationary point *x of a vector field v is defined by*

$$v(x) = 0.$$

To classify critical points $x \in \mathbb{R}^n$ in vector fields, the field is linearized at x as

$$v(x) = J \cdot x + b$$

where J is the Jacobian of v at x. Contrary to the Hessian of scalar fields, the Jacobian can be asymmetric and its eigenvalues may be complex-valued due to rotation. This rotation can create critical points whose neighborhood describes a rotating motion such as *rotating stars* and closed trajectories.

Definition 3.36 (Cycle) *A streamline $x(t)$ is a* cycle *or a closed trajectory if there exists a $\Delta t \in \mathbb{R}$ such that*

$$x(t + b\Delta t) = x(t) \quad \forall b \in \mathbb{N}.$$

These cycles are a special type of streamline limit set, which is not present in gradient fields. In vector field topology attractors are called *sinks*, repulsors are called *sources*. Ascending and descending manifolds are known as *unstable* and *stable manifolds* in general vector fields.

A topological graph of the field is obtained by connecting saddles with critical points. The resulting separating streamlines partition the field into regions of similar flow behavior.

Definition 3.37 (Separatrices) *A* separatrix *is a feature that separates different regions of homogeneous flow behavior.*

These separatrices are integral flow features and emerge from saddles, where they are locally tangential to the eigenvectors of the linearized flow field. From these saddle positions, they are integrated as streamlines from positions offset along the direction of major and minor eigenvectors and are tangential to flow direction leading to a sink, source, or leaving the domain. Thus, separatrices are the stable and unstable manifolds of saddle points. In a scalar field these lines follow the direction of maximal ascent/descent and can give a notion of height ridge and valley locations.

Saddles represent nodes of the topological graph located inside a vector field. However, there are situations where vector field behavior suggests the existence of separation processes outside of the field domain. A source of separatrices that captures this behavior and is located on field boundaries are separation and attachment structures.

Definition 3.38 (Separation and Attachment Lines) Separation lines *and* Attachment lines *are locations at field boundaries, where flow separates from or attaches to the boundary.*

Mathematically speaking, separation and attachment lines can be described for the projection of the flow field onto the boundary. These locations p are defined as positions, where the major or minor eigenvector of the Jacobian J of the projected flow field $v : \Omega \subset \mathbb{R}^n \to \mathbb{R}^n$ is orthogonal to the projected flow direction:

$$v(p)|_{\partial\Omega} \cdot e_{\{max,min\}}(J(p)|_{\partial\Omega}) = 0$$

Like saddle points, these separation and attachment lines serve as starting points for separatrix integration. Separatrices end at these positions, whereas neighboring integral flow lines and surfaces are deflected along the boundary.

3.5.2.2. Instationary Vector Fields

Concepts of scalar field and stationary vector field topology cannot be directly transferred to time-dependent vector fields. While these methods can convey a picture of instantaneous topology of a time-varying vector field, this graph does not correlate with the limit behavior of time-varying flow. There are however ways to identify hyperbolic structures in time-varying vector fields, which are presented in the following but do generally not produce topological graphs. *Lagrangian Coherent Structures* (LCS) represent regions in a time-varying vector field, where particle trajectories show strong converging or diverging behavior. LCS are height ridges in the *Finite Time Lyapunov Exponent* (FTLE) field [HY00].

Definition 3.39 (Finite Time Lyapunov Exponent) *The* Finite Time Lyapunov Exponent *field is a scalar-valued field derived from the velocity field, by computing a flow map $\Phi_{t_0}^{t}(x)$ that maps flow particles x at time t_0 to their respective positions in timestep t. The FTLE value at x then corresponds to the scaled logarithm of the maximal singular value of $\nabla\Phi_{t_0}^{t}(x)$*

$$\sigma_{t_0}^{\delta}(x) = \frac{1}{|\delta|} ln(\sqrt{\lambda_{max}(\nabla\Phi_{t_0}^{t_0+\delta}(x)^T \cdot \nabla\Phi_{t_0}^{t_0+\delta}(x))})$$

where δ represents maximal advection time.

Consequently, the FTLE field models exponential deformation of a particle neighborhood for a given advection time δ. High values in the forward FTLE field ($\delta > 0$) therefore indicate exponential divergence in the flow field, high values in the backward FTLE field ($\delta < 0$) express (forward) convergence of flow. From this scalar field, LCS can be extracted as height ridges. Flat regions in the FTLE field may be interpreted as regions with homogeneous flow behavior.

3.5.3. Tensor Fields

Tensor field topology is concerned with the analysis of eigenvector fields. More particular, one is interested in the behavior of hyperstreamlines and tensor-lines.

Definition 3.40 (Hyperstreamline) *A* hyperstreamline *of a tensor field is a streamline that is tangential to one of the fields eigenvector fields.*

Examination of the behavior of hyperstreamlines reveals regions in fields, where such lines cross. These features (also known as *umbilics*) form the skeleton of tensor field topology.

Definition 3.41 (Degenerate Points) *A point in a tensor field, where at least two of the eigenvalues are identical is called a* degenerate point*.*

In three-dimensional space, these features form stable *degenerate lines* rather than points if the field is partially isotropic [ZP04]. Topology of the complete tensor field is then obtained by connecting degenerate features with separatrices consisting of hyper-streamlines or hyperstreamsurfaces.

3.5.4. Numerical Methods

In non-analytic fields, location of critical points is performed numerically by local root finding methods such as Newton-Iteration. In complex fields, this is computationally expensive, with the additional challenge of guaranteeing to find all critical points in scattered data sets. In gridded fields, root finding may be performed cell-wise and (with appropriate interpolants) analytically.

The construction of a Morse-Smale complex in a normalized discrete scalar field f can be achieved without the explicit extraction of critical points by performing watershed segmentation of f and $1-f$ [ČDFP05]. The intersection of the resulting watershed segmentation corresponds to an approximation of the discrete Morse-Smale complex. However, a necessary condition for this approximation to model the Morse-Smale complex is that f is a Morse function. This does not generally hold for discrete scalar fields. Fortunately, suitable pre-processing techniques can enforce this condition [Ede01].

Separation and attachment lines are commonly extracted on cell-wise projections of the field on the tessellated boundary mesh of the data set. For this matter, lines are approximated by segments created from intersections of these lines with edges of boundary mesh elements [KHL99].

A major challenge is the detection and extraction of closed trajectories in flow fields [WS01]. The detection of these structures requires highly accurate tracing of trajectories as provided by high-order adaptive numerical integration schemes.

Separatrices that emerge from saddles are by definition integrated in the direction of flow convergence. This guarantees accurate approximation of separatrices even in cases with low-order integration methods. The same holds for separatrices that emerge from separation lines.

4. Complex Valued Scalar Fields - Mesh-Free Valley Surfaces

This chapter is concerned with the extraction of valleys and ridges as generalized extrema in complex-valued three-dimensional scalar fields. In the following sections we present a new region-growing based approach to the mesh-less extraction of adaptive non-manifold valley and ridge surfaces that overcomes limitations of previous approaches by decoupling extraction and triangulation of the surface. Our algorithm allows the resulting valley surface skeleton to be extracted as a connected structure rather than set of disconnected surfaces or point sets. As our algorithm is inherently mesh-free and curvature adaptive, it is suitable for surface construction in fields with an arbitrary neighborhood structure. To perform surface extraction, we analyze behavior of the derived Hessian and gradient fields near degeneracies, which allows application of the maximal convexity ridge definition. The decoupling of the extraction from the triangulation step as well as the absence of a computational mesh allows a less constrained and more regular surface construction than provided by previous methods.

The extraction of ridge and valley structures (also termed *creases* in the following) [Ebe96] has a long history in the field of image processing, where ridges in the scalar intensity field represent extremal features such as highlights of an image. However, crease line and surface extraction can be applied to a wide area of visualization problems ranging from such topics as three-dimensional medical imaging to time-dependent vector field analysis, where the loci of these generalized local minima and maxima facilitate topological analysis of field structures in the context of FTLE ridges.

As an application for insightful visualization with valley surfaces, we choose a low frequency acoustics simulation. The visualization results presented in this chapter are a step towards understanding the nature of FEM solutions to the wave equation of simulated acoustics, where regions of locally minimal pressure represent wave nodes and regions of sound cancellation. Therefore, the application of the presented methods focuses on scalar pressure fields that are eigensolutions of an FE model used in low frequency acoustics simulations. Due to its periodic nature, the time-varying pressure field is represented by complex numbers encompassing amplitude and phase shift, allowing the valley surface method to visualize both stationary and time-varying topology of wave nodes in this field. This provides an expressive visualization of wave node and anti-node structures in simulated acoustics. As complex fields encode multiple time-varying scalar fields, namely amplitude and phase fields, feature extraction and visualization in such fields is directly related to multi-field processing.

Our method contributes to the visualization area by presenting a novel region-growing technique for the mesh-free extraction of curvature adaptive crease surfaces. We propose a three-dimensional seeding method that is not constrained by triangulation needs and allows easy control of surface resolution. For surface convergence, we analyze an analytical alternative to Newton iteration techniques. Furthermore, we introduce a technique to connect maximal convexity ridges by slope line integration to obtain non-manifold minimum structures. Researchers from the field of acoustics benefit from the insights gained by applying our extraction and visualization techniques to sound simulation data sets.

The second section of this chapter gives an overview of related work in the fields of crease surface extraction and acoustics visualization. In Section 4.3, we detail the mathematical foundations of crease surfaces before proposing our novel extraction algorithm in Section 4.4. Section 4.5 introduces a novel application of ridge line extraction, namely simulated acoustics, providing background knowledge of the numerical examples shown in Section 4.6. Section 4.7 summarizes this chapter and gives a short outlook on possible future work.

4.1. Related Work

In this section we present related work from the areas of crease extraction and sound visualization.

4.1.1. Crease Extraction

Related work in the context of the crease extraction part of this work comes from two different fields. The first topic is concerned with iso-surface extraction techniques in arbitrary scalar fields. Work of the second field is focused on crease line and surface extraction and its applications.

A first approach to mesh-free implicit surface reconstruction was introduced by Hilton et al. in 1996 [HSIW96]. Their *Marching Triangles* algorithm is a region-growing based approach to the construction of Delaunay triangulations of iso-surfaces. More recent work such as the methods by Akkouche et al. [AG01] and Xi et al. [XD08] make enhancements to this technique by creating curvature adaptive semi-regular meshes of such manifold iso-surfaces. Related work using partial differential equations to extract particle based isosurfaces from unstructured point sets is given by Rosenthal et al. [RL08]. A similar approach that focuses on an iso-surface curvature measure, heuristic edge lengths and front tracking was presented by Araujo et al. in 2004 [dAJ04]. In 2007 Meyer et al. [MNKW07] proposed the use of particle systems for isosurface extraction in reference space. Their energy-based approach allows to sample isosurfaces adaptively according to surface measures such as curvature. All of these methods have in common that they are limited to manifold iso-surface extraction and produce disconnected point sets or have to impose certain constraints on surface extraction resulting from desired properties of the final surface triangulation.

Work on crease structure extraction has conventionally focused on manifold reconstruction in uniform, hierarchical or structured grids. A well-known method is the "Marching Ridges" approach by Furst et al. [FP98], which is a Marching Cubes related algorithm to extract height ridges in image data. While a number of other techniques focus on the extraction of creases in images [Ebe96], more general approaches have been published recently. Sadlo et al. [SP07] present an approach with adaptive grid refinement for the visualization of Lagrangian Coherent Structures of unsteady vector fields. Another method by Kindlmann et al. [KTW06] introduces an alternative scheme to orient eigenvectors of the Hessian matrices of scalar fields to extract creases in diffusion tensor MRI. A first approach to cover non-manifold areas of crease structures in diffusion tensor MRI is given by Schultz et al. [STS10], which improves over standard marching cubes extraction techniques by reconstructing smooth surface boundaries. Another grid-based approach that focuses on the open nature of ridge surfaces was recently introduced by Li et al. [LLP$^+$10]. In grid-based methods the quality of the triangulation is heavily dependent on the chosen grid and not curvature dependent, reducing the overall accuracy and condition of the extracted surface mesh.

While these techniques aim at obtaining surface triangulations, newer work [KESW09] has brought the well-known concepts of energy based particle distributions and scale space [Lin96, Dam99] to the field of three-dimensional ridge visualization. Others [BT10] have recently proposed methods for GPU-based extraction and visualization of (optimal-scale) ridges. While such GPU-based techniques are commonly limited in accuracy, and restricted to certain data structures, they allow for interactive framerates. Methods for the detection and visualization of ridge and valley lines on surface meshes [KK06, OBS04] are related to crease extraction in scalar fields, while serving a different visualization purpose and relying on fundamentally different extraction techniques.

Our work uses a continuous polynomial field approximation to generate curvature adaptive sets of points that can be meshed due to neighborhood information obtained during surface extraction. The resulting mesh can be used as basis for further visualization techniques such as shading, texturing, and volume rendering. Additionally, we propose a method to include non-manifold regions into the surface extraction process and emphasize the importance of valley surface extraction to a novel application area, namely visualization of complex valued acoustic pressure fields.

4.1.2. Sound Visualization

Sound visualization techniques processing simulation or real world measured data mostly focus on wave propagation or scalar field visualization of values such as pressure or other acoustic metrics. Data from acoustics simulations is commonly available as continuous function representing wave or sound propagation or is computed and stored at individual listener positions.

Visualization techniques for wave propagation are proposed in [YST02, PR05c, PR05b] for two- and three-dimensional sound fields. In [BDM$^+$05] sound propagation is visualized by means of sound particle tracing. Lauterbach et al. [LCM07] make use of frustum tracing to visualize and simulate sound propagation in game-like room environments. The work by Tokita et al. [TY05] uses particle displacements on a three-dimensional grid as basic visualization technique. In [PL03] Pulkki et al. present an approach visualizing simulated edge diffraction with the image source method. Funkhouser et al. [FCE$^+$98] use visualization of sources and listeners as well as sound paths for analysis and evaluation of their acoustic modeling method. Khoury et al. [KFW98] represent the sound pressure levels inside the room on color mapped slicing planes and analyze wave front propagation with isosurfacing techniques. However, without the use of transparency or volume rendering techniques these methods are limited in three dimensions due to visual occlusion. Merimaa et al. [MLPK01] present a visualization of directional room responses using two-dimensional plots depicting intensity and propagation direction. Omoto et al. [OU04] present a circle-based visualization metaphor that carries information about intensity and direction of incoming sound waves for different receiver positions. Weyna [Wey05b, Wey05a] make use of vector field integration methods to analyze acoustic flow fields in the vicinity of sound obstacles. Stettner et al. [SG89] use icons and ray diagrams to visualize sound metrics such as overall sound strength and clarity and definition. Monks et al. [MOD00] show sound strengths for different source types by color mapping listener positions and room geometry.

For a complete overview of existing sound visualization methods we refer to [LN06, Dei08].

Our intention is to represent the topology of minimum structures of the room response and visualize the results in complex valued scalar fields using our novel valley surface reconstruction algorithm. This allows clear statements about wave node properties and development. To our knowledge no such method has been introduced before.

4.2. Crease Surface Theory

Research has produced a number of different definitions and notions of local n-dimensional minima, as regions with locally minimal amplitude are important for a vast topic of applications from different fields. In this work, we make use of two related ridge definitions. The first definition has its origins in image processing and is given in [Ebe96]. Its features are known as *maximal convexity ridges* as they correspond to locations of maximal value along directions of maximal curvature or convexity. This standard mathematical definition of feature points x on crease structures of a twice continuously differentiable three-dimensional scalar field $f : \Omega \subseteq \mathbb{R}^3 \to \mathbb{R}$ with gradient ∇f and Hessian H_f is given in the following. As this work focuses on the visualization of minimum structures, we refer to valley structures in the following instead of creases in general. All definitions and statements hold for ridges as well, which are the valleys of the negated scalar field. Let $\lambda_1 < \lambda_2 < \lambda_3$ be eigenvalues of H_f at x with corresponding eigenvectors e_1, e_2, e_3, then x is a point on a k dimensional valley, iff $\lambda_{k+1} > 0$ and:

$$\nabla f(x) \cdot e_i = 0, \; \forall i > k. \tag{4.1}$$

Geometrically speaking, such points lie on structures where the gradient of a the scalar field is orthogonal to the field's direction of maximum curvature. This fact about the behavior of two vector fields has been employed in [PR99], where the parallel vectors operator was used for extraction of such features. Equation (4.1) classifies extrema, crease lines and crease surfaces as 0D, 1D, and 2D creases. This definition does not allow branching of minimum structures.

This notion of generalized minima makes it obvious, why traditional methods for iso-surface construction fail to extract creases. Firstly, there is no inherent change in sign on opposing sides of the structure, as eigenvector fields and valley surfaces generally lack orientation. Secondly, points on the same crease may vary greatly in function value. There are however ways to locally orient eigenvectors to find sign changes on edges of cells that do not contain a Hessian degeneracy (two equal eigenvalues) [KTW06, FP98]. Combinatorical methods allow the handling of degeneracies as well [STS10].

The second type of minimum structure definition is wide-spread in the field of vector field topology [HH91] and terrain analysis and is based on slope- and separation line extraction. As stated in Chapter 3, edges in the Morse-Smale complex of a scalar field can be interpreted as ridge or valley structures. Compared to watershed or slope line segmentation, maximal convexity definitions rely on local properties of the field only and avoid the expensive steps of global explicit extrema detection and streamline or stream surface computation. An informative analysis regarding the differences between maximal convexity ridges, watersheds, and slope line algorithms can be found in [KvD93, SWTH07, PS08].

4.3. Valley Surface Construction

Given the definition of maximal convexity ridges and slope lines, we define the following steps for mesh-free valley surface extraction.

4.3.1. Extraction

The basic valley surface extraction algorithm relies on the maximum convexity ridge definition and performs the following steps:

1. Place initial surface point
2. Seed neighboring points
3. Converge to valley surface
4. Merge points
5. Repeat from 2, until no free points left

These steps are extended as described in Section 4.3.2 to allow for non-manifold structures in valley surfaces. In most of the steps, an approximation of the scalar field along with its first and second derivatives is needed. We compute this data by locally fitting a trivariate polynomial to the scalar data using the *Moving Least Squares* technique, see Appendix A. Further details on the extraction steps are given in the following sections.

4.3.1.1. Pre-Processing of the Data Set

Scattered scalar fields are pre-processed to facilitate fast local field approximations. We construct an uniform grid with a cell size corresponding to approximately 2.5 times the average point distance to speed up the process of data point location. This basic grid resolution has shown to provide a good balance between empty and overly filled cells in data sets with moderate changes in point distance. We choose a similar value for the radius of our MLS kernel during continuous field reconstruction, thus defining a basic scale for extrema detection. If a larger scale is desired due to low point density or during convexity approximation, this radius is increased and the number of adjacent cells covered by the support function used for approximation is enlarged. After this pre-processing step, valley surface extraction may start.

4.3.1.2. Initialization

As a starting point for region growing, we find an arbitrary surface point in the domain of the scalar field f by performing Newton iteration along the main direction of curvature e_3. Iterations are started at centroids x of cells.

$$x_{i+1} = x_i - e_3 \frac{\nabla f \cdot e_3}{\lambda_3} \tag{4.2}$$

As shown in [Ebe96], the normal n of a maximal convexity ridge is computed as

$$n_i = \sum_j e_{3_j} \frac{\partial f}{\partial x_j \partial x_i} + \sum_j \frac{\partial e_{3_j}}{\partial x_i} \frac{\partial f}{\partial x_j} \tag{4.3}$$

where $\frac{\partial e_{3_j}}{\partial x_i}$ effectively requires third derivatives of f. Since the computation of third derivatives requires approximation of the scalar field by cubic polynomials or the use of finite differences, our convergence scheme uses e_3 as rough approximation of the normal and only falls back to more accurate finite-difference normal computation in cases of divergence during seeding (cf. 4.3.1.4). The initialization phase is completed as soon as a first point of convergence was reached. This first point serves as initial seed point of the surface and keeps e_3 as initial normal approximation. It acts as the initial parent to new points during the seeding step.

Note that the valley surface extraction algorithm is restarted after the extraction of a complete surface to find other disconnected surface structures in the data set, as detailed in Section 4.3.1.6.

4.3.1.3. Point Seeding

Region growing is performed by point seeding. We choose a point p of the surface that has not yet been source of seeding and create new neighboring child points on equally distributed positions on a planar ellipse E around p

$$E(r_1, r_2, t) = \begin{pmatrix} r_1 cos(t) \\ r_2 sin(t) \end{pmatrix}$$

where $r_1, r_2 \in \mathbb{R}$ correspond to radii in direction of the major and minor axis of the ellipse, as shown in Fig. 4.1. Throughout the rest of this paper, $t \in [0, 2\pi]$ is sampled by $\Delta t = \frac{1}{3}\pi$, and axes of the ellipsoid are oriented such that one axis points towards the parent point. This ellipsoid lies in the plane orthogonal to the normal of p and represents a first linear approximation of the valley surface. Axes lengths of the ellipsoid are determined by an approximation of valley surface curvature. During the first seeding step of a surface, axes lengths are set to one quarter of the grid cell size, as determined during loading of the data set.

We approximate the curvature at an arbitrary point p on the surface by analyzing normal deviation between p and its parent p_0. For that matter, we compute

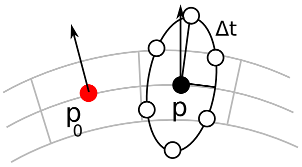


Figure 4.1.: Seeding on an ellipsoid around point p. A normal deviation between p and its parent (red) leads to a scaled ellipsoid.

the radius of the circle passing through p, p_0 and p', where p' is created by mirroring the edge $e = (p_0, p)$ through the normal n_{p_0} as seen in Figure 4.2a. This form of curvature approximation was used in [MS92] and has shown to be a valid and robust way of obtaining a fast notion of surface curvature along e. Correspondingly, we compute the surface curvature along an edge $e_\perp$ orthogonal to e, where $e_\perp$ is obtained by projection of $n \times e$ onto the neighborhood triangle-fan (see Figure 4.2b). According to Thales' theorem, this radius of curvature computes as:

$$r = \frac{1}{|curv_e|} = \left|\frac{(e^T \cdot e)}{2.0 \cdot (e^T \cdot n_{p_0})}\right| \tag{4.4}$$

Axes of the new seed ellipsoid used by p are scaled by a fraction of these two curvature estimates to provide bi-directional surface adaptivity. Extremal edge lengths are constrained to the magnitude of the data set cell size to avoid intensive super- or sub-sampling. Numerical examples of the impact of curvature on edge lengths is given in the results section of this chapter.

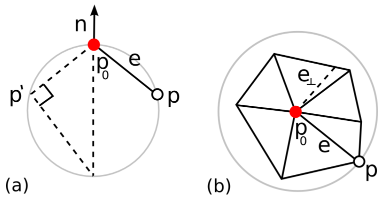


Figure 4.2.: Curvature approximation is performed on a two-dimensional projection of the surface. Curvature measure along e viewed from the side (a) and bi-directional approximation for full surface adaptivity viewed from above (b).

4.3.1.4. Converging to the Valley Surface

Points p placed on planar approximations of the surface during seeding do generally not lie on the real minimum surface. We therefore project these points to locations on the true surface by using either one of the following two methods:

(i) a standard Newton iteration along the direction of main curvature according to (4.2), or

(ii) by moving to the minimum of a quadratic trivariate polynomial that approximates the scalar field.

The second method has a number of advantages over the first. Firstly, it makes use of continuous polynomial field approximation and secondly, it generally reduces the number of iterations performed until a point of convergence is reached. The minimum of a quadratic trivariate polynomial at p can be found analytically. Let

$$h(x, y, z) = c \cdot (1 \quad x \quad y \quad z \quad x^2 \quad xy \quad y^2 \quad yz \quad z^2 \quad xz)^T \tag{4.5}$$

be such a polynomial approximating the scalar field with coefficient vector $c \in \mathbb{R}^{10}$, as obtained by polynomial approximation such as MLS. By restricting h to positions on the line $g(t) = p + t \cdot e_3$ with $t \in \mathbb{R}$, we obtain the analytically differentiable univariate polynomial

$$(h \circ g)(t) = c \cdot (1 \quad g(t)_x \quad g(t)_y \quad \ldots \quad g(t)_{xz})^T \tag{4.6}$$

whose minimum lies on an approximation of the valley surface. Generally, this scheme converges faster than the common Newton scheme, which is based on linear approximations of the surface normal and field gradient. Divergence signals a boundary of the valley surface.

If the starting points or approximations are chosen unwisely, these convergence schemes have the common problems of local minima and divergence. This is an additional motivation for the use of appropriate adaptive seeding step sizes in areas with high curvature. We consider a point to have reached the valley surface if

$$\left| \frac{\nabla f^T}{\|f\|} \cdot e_3 \right| < \epsilon$$

indicates that the enclosed angle is close to zero, where ϵ typically takes values of the order $10^{-10} - 10^{-4}$. In certain situations, points diverge or are otherwise invalid, especially when leaving the domain of the data set or crossing one of the boundaries described in 4.3.2. In this case, the corresponding point is reinitialized with a position half way to its parent and the used iteration scheme is restarted with a finite-difference approximation of the surface normal. This is repeated until the point converges, or the distance to its parent falls below a pre-defined minimum. This procedure guarantees smooth reconstruction of surface boundaries. Once all points have converged to the valley surface, we update normal data of the parent point based on child positions to reflect the expected surface (mesh) normal rather than the direction of main curvature.

4.3.1.5. Merging of Points

The current surface point set contains a graph implicitly defined by the parent-child neighborhood and the neighborhood relation between neighboring points on seeding ellipsoids. To maintain a correct neighborhood graph with a curvature adaptive point density, we merge new points with existing ones if the distance to the closest existing point falls below a threshold θ. This threshold is empirically chosen to be approximately 20% less than the distance to its parent. The more recent point is abandoned and its parent connected to its merge partner as depicted in Figure 4.3.

These region growing steps are repeated for each new point that was not merged into the existing point structure. This surface growing approach avoids the need of front tracking and splitting as well as detecting triangle overlaps and provides a method for triangulation independent surface generation. While a non curvature adaptive approach based on Newton iteration on a continuous scalar field was presented in [KESW09], our approach avoids expensive inter-particle energy function minimization, guides particle starting positions along an approximation of the valley surface and provides means for surface meshing.

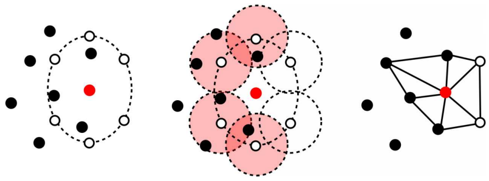


Figure 4.3.: Seeding and merging after convergence with new neighborhood structure.

4.3.1.6. Independent Surfaces

Once growing of a distinct valley surface is completed, finding independent disconnected valley surfaces in other parts of the data set requires the search for new seed points. As the highest reasonable reconstruction resolution is defined by point densities in the data set, we limit search for new valley surfaces to cells in the grid discretization that were not covered by existing valley surfaces. Our tests have shown that this method recovers all valley surface structures in data sets with moderate variety in point densities. If variation in point density is high, these searches have to be performed in an according number of smaller cells.

4.3.1.7. Progressive Surface Generation

In contrast to other particle-based crease extraction methods, the dynamic and local nature of our surface growing approach is suitable for interactive and progressive surface extraction. Several steps of the aforementioned algorithm qualify for optional user-input:

- Selection of a locally constrained seed region
- Selective surface growing
- Local surface refinement

Given the appropriate interaction tools, selective surface growing reduces to selection of existing surface points as new seed points and specification of a maximal growth radius. The existing surface point graph does not require distinction between user-selected and automatic seed points, as new surface structures are automatically merged into the existing complex. The same holds for progressive surface refinement, where selection of surface parts and level of curvature adaptivity facilitates automatic refinement of the surface by seeding new points around existing surface particles. Section 4.5.2 demonstrates the capability of our algorithm to handle user-guided surface growing and refinement.

4.3.2. Non-Manifold Regions

Valley surfaces produced by the definitions and steps presented in the previous sections are maximal convexity ridges and are as such (disconnected) non-branching surfaces. Points p on boundaries of valley surface defined by (4.1) are characterized by one of the following properties:

(i) p is part of the data set boundary

(ii) $\lambda_3(p) = 0$ ($H_f(p)$ is singular)

(iii) $\lambda_2(p) = \lambda_3(p)$ ((partial) umbilic point)

Boundaries of the second type are caused by a change in sign of main curvature. The case of a singular Hessian, where $\lambda_3 = 0$, can in fact indicate a minimum rather than a local plateau that is missed by the compact support of common scattered data approximation techniques. If our surface reaches a boundary of this type during region-growing, such an event can be detected by increasing the radius of the support function chosen for field approximation, thus giving a notion of a different scale. This is continued until the maximal curvature value is non-zero.

While the last type of boundary is a valid surface border according to (4.1), it prevents extraction of general minimum regions, where $\lambda_2 > 0$. Curvature line configurations of these (partial) umbilic points [DH94] can be divided into three classes, namely *stars* (trisector), *monstars* and *lemons* (wedges) [LHL+08]. From this classification, the trisector type is of main interest to the behavior of

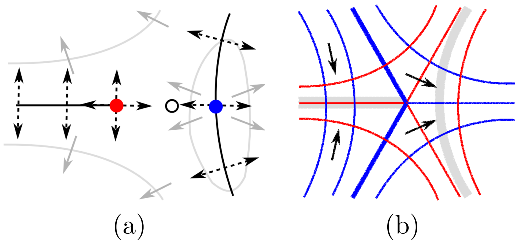


Figure 4.4.: (a) Partial umbilic point where $\lambda_2 = \lambda_3$ (red) and point on opposing valley line (blue), where $\nabla f \perp e_2$ and $\nabla f \perp e_3$. Directions of maximum curvature and surfaces are shown in black, gradient vectors and iso lines in gray. (b) Maximal (blue) and minimal (red) curvature lines around umbilic point. The star type of the umbilic point separates convergence regions for the two valley lines shown in gray, as indicated by the bold principal curvature lines. Black arrows represent convergence directions.

our valley extraction algorithm. Figure 4.4a depicts a two-dimensional example of gradient and valley line behavior in the vicinity of a star-type umbilic point. The according star configuration with principal curvature lines and convergence regions is shown in Figure 4.4b. Curvature line separation causes the maximal convexity ridge to the left to end at a position, where $\lambda_2 = \lambda_3$.

A more general definition of 2D minimal structures [FKMP97] in three-dimensional space that includes maximal convexity ridges requires that a point is a local minimum with respect to at least one direction to be part of a general valley surface. According to this definition the outlined point in Figure 4.4a, which is placed past the umbilic point, continues to be a minimum with respect to the vertical direction but fails to be a maximal convexity ridge point, as the direction of main curvature is now horizontal. Extraction of points on these general minimum surfaces conveys important information about the connectivity of (maximal convexity) minima in the data set. Our approach attempts to resolve these situations by incorporating these points into the extracted valley surface structure, comparable to connector structures defined in [FKMP97]. The resulting structures allow for (branching) non-manifold valley surfaces.

4.3.2.1. Surface Branching

Regions where valley surface branching may occur are either valley lines or loci of partial umbilic points (see Figure 4.4a). During surface growing, we approximate valley lines by finding points on the valley surface with the additional constraints that $e_2 > 0$ and $\nabla f \perp e_2$. To converge to crease lines, we restrict the Newton iteration step to a plane orthogonal to the direction of minimal curvature:

$$x_{i+1} = x_i - \nabla f_p \frac{||\nabla f_p||}{(H_f \cdot \nabla f_p) \cdot \nabla f_p} \tag{4.7}$$

where ∇f_p is the gradient projected to the plane of convergence. Again, this can sped up in the case of quadratic approximation, where the field minimum on this plane spanned by e_2 and e_3 can be found analytically. Partial umbilic points on the other hand are identified as locations on the surface boundary, where $\frac{\lambda_2}{\lambda_3} > 1 - \delta$ and a change in main curvature direction was detected during convergence. An numerical procedure for finding these stable degenerate lines in grid-based fields is presented in [ZP04]. After identification of these branching locations, the remaining challenge consists of correct connection of partial umbilics and valley lines. To achieve this goal, we distinguish two cases:

If $\lambda_2 >> 0$ close to the degenerate line, we extract the connector structure by replacing λ_3 in (4.2) by λ_2. The resulting connector surface merges corresponding crease lines with degenerate lines. This extraction technique is however numerically not reliable in regions, where the field is flat in direction of medium curvature, as this causes deviations in approximated medium curvature direction.

Crease lines on crease surfaces are related to separation and attachment lines on general surfaces in vector fields. The criterion by Kenwright et al. [KHL99] defines separation and attachment lines as locations, where the projected flow field on a mesh is parallel to one of the projected Jacobian's real eigenvectors. At points on valley lines the gradient field is parallel to to the direction corresponding to the minimal eigenvector of the Hessian of the scalar field. Therefore, such a line feature on a crease surface represents a location, where gradient flow separates or attaches to a valley surface. The existence of such a line feature suggests the presence of a neighboring ridge structure in the direction of gradient flow as shown in 4.4a. As points on these line features are sources of the gradient field's two-dimensional projection onto a plane orthogonal to the direction of minimal curvature, they represent important features in the context of vector field topology based separation line extraction.

During surface growing, one of two situations may occur: the surface may approach the branching region and stop growing, as new child points fail to converge due to a degenerate line boundary (surface on the left of Fig. 4.4a). Alternatively, the surface may continue growing tangentially to the minimal and medium curvature directions and cross the junction-like region.

If the first case occurs, the inverse gradient direction points towards the branching region and describes the direction of a flow separation line. We therefore continue stepping along the negative gradient direction, as long as $\lambda_2 > 0$ until we meet the maximal convexity ridge at a crease line point. We constrain this gradient descent to a plane orthogonal to the direction of minimal curvature, on which the projected vector field degenerates to a source point, where the crease line intersects the projection plane. We merge the created start and end positions of these connector lines into the existing valley structure and update point neighborhood relations accordingly.

The second case indicates a nearby valley surface and is used to seed a set of points in the neighborhood to ensure that all nearby structures are recovered by the extraction algorithm. Both situations are shown in Figure 4.5. Ascending gradient lines show divergence due to flow separation and are therefore not suitable for surface merging.

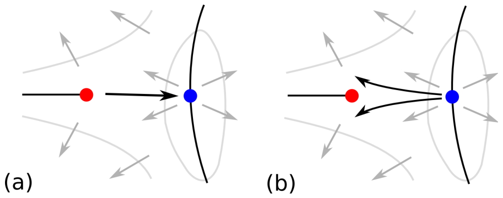


Figure 4.5.: In non-manifold regions gradient descend (a) and gradient ascent (b) is performed. The former method extracts minimal points along slope-lines in non-manifold regions, as slope lines show converging behavior. The latter method is used to find adjacent valley surfaces and needs an additional convergence step to meet the valley surface on the left.

There exists work on vector field topology that uses stream surfaces to produce similar topological structures. However, we find that flow based surface integration is more suitable in the context of true vector fields than scalar field visualization, where gradient directions close to separation structures often vary substantially, leading to numerically unstable integration of many small gradient stream surface parts. Furthermore, stream surface integration has the disadvantage of favoring the gradient direction, which is unsuitable in the context of crease surface extraction. Our approach uses a constrained gradient based surface integration in areas of ambiguous convexity only, as the gradient has a distinct and consistent direction in these regions.

4.3.3. Triangulation

After surface extraction has created a point cloud representing a connected surface graph, this neighborhood relation keeps important surface connectivity information for triangulation. Using this connectivity information, we triangulate the final surface by creating local Delaunay triangles. This procedure follows the principle of standard Delaunay meshing techniques on the manifold surface parts. Hereby edge swapping has to ensure that crease lines form edges in the triangulation.

After manifold triangulation, we identify opposing crease- and degenerate lines. Starting with a crease point, we find its nearest neighbor in the surface graph on the edge of a triangulated part, i.e. a point close to a degenerate line. From there, we create two point lists by moving in forward and backward direction on the crease and opposing degenerate line, placing neighboring surface points in the appropriate list for the crease- or its opposing line. The ribbon bound by the two line sequences defined by these lists covers the corresponding non-manifold region. This ribbon is triangulated using standard ribbon-triangulation techniques by alternately stepping along the lines and creating triangles based on a quality criterion such as minimal triangle edge length, see (Fig. 4.6 and [Hul92]). Triangle creation for a ribbon stops as the ribbon ends or meets its front in case of a cyclic ribbon.

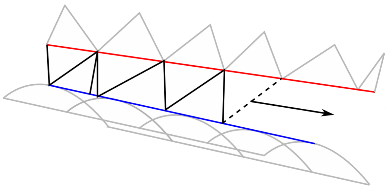


Figure 4.6.: Crease line (blue) and degenerate line (red) are connected during the ribbon triangulation step.

4.4. Application

In the field of simulated acoustics, valley surfaces contain important information about the structure of the field, wave behavior as well as general sound properties. In standing waves, valley surfaces describe node regions, i.e. areas of sound cancellation where the amplitude value is virtually zero. Valley surfaces in moving waves are able to visualize time-varying regions of minimal amplitude, as well as areas where the maximal amplitude of an oscillating system possesses a minimum, as demonstrated in Section 4.5. These capabilities allow the scientific analysis and optimization of acoustic room properties and make valley surfaces a powerful tool for acoustics visualization.

Sound simulations provide domain experts with numerical representations of sound attributes such as sound pressure, amplitude and phase that allow an analysis of acoustic properties of the underlying geometry and medium. This data is either evaluated at specific listener positions, or given at a comparatively dense set of evaluation points that can be used to reconstruct a continuous field representation. Important input parameters for such simulations are geometric shapes, reflection/absorption coefficients for participating materials and frequency, amplitude and location information for the simulated sound sources. Wave-based acoustics deals with the numerical solution of the wave equation. For sufficiently low frequencies, simulations based on finite element discretization techniques can be applied to real world examples, as a comparatively low grid resolution does already provide aliasing free accurate sampling of waves. Higher frequencies increase computational complexity significantly and are therefore not suited for fast FEM calculations. In this context frequencies are classified as low with respect to a given environment if the product of characteristic domain extensions/lengths and frequency is sufficiently low. Consequently, low frequencies in a concert-hall sized environments commonly fall below the hearing threshold of human ears, leading to the fact that low frequency analysis in acoustics is usually focused on studio-sized rooms.

A number of different discretization and modeling techniques such as *Functional Transform Method* (FTM), *Finite Difference Time Domain* (FDTD), and *Finite Element Method* (FEM) may be used to solve the wave equations for either one specific, or a whole spectrum of frequencies at different parts of the data set domain. FTM [TR03] is originally limited to simple geometry, but allows dense computation in the parameter domain. FTM was extended to more complex geometry and applied to sound wave propagation by Petrausch et al. [PR05c, PR05a] and [RNM09], where an analytical solution of the Partial Differential Equations is computed on the GPU. Finite differences simplify the solution of the time dependent wave equation in FDTD methods [Bot95] and yield sampled pressure and velocity fields for a range of different frequencies. Sound pressure values at positions throughout a volume with high geometrical complexity may be obtained by FEM [Bra03, Ihl98], which solve ordinary differential equations at grid points, depending on absorption coefficients of different materials in the data set. The high number of unknowns in these systems of ODEs requires state-space reduction models [DBM$^+$06] to provide a solution in real time.

We use output of an FEM simulation that provides us with a complex valued sound pressure field encoding amplitude and phase shift values at points of an irregularly sampled data set for a given low frequency wave.

4.5. Results

4.5.1. Application to Sound Visualization

The first test geometry for our low frequency visualizations is a small completely tiled reference room with two doors and a radiator on one side. In Figure 4.7 the room geometry and the finite element mesh used for the simulation is shown. The sound source in this case is the membrane of the loudspeaker which is visible in the corner of the room. Complex pressure values are available at approximately 21000 positions throughout the room. The second test room is an artificial L-shaped room with a sound source centered at its shorter end. It consists of about 220000 sample points. The data sets are virtually noise free and require no pre-smoothing to reduce high frequency clutter during extrema extraction as is common in the field of image processing. The results shown in the following map scalar quantities for color mapping purposes to hue in HSV space.

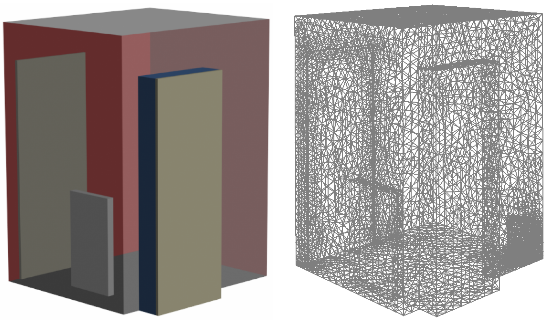


Figure 4.7.: Geometry and FEM mesh of the simulated room.

Amplitude values of the complex pressure field give a stationary view of the sound pressure distribution in the rooms. Hereby, the amplitude value

$$A(x) = \sqrt{im(x)^2 + re(x)^2}$$

at a given point x corresponds to the maximal amplitude of the oscillating system at that position. For stationary waves, position of the node structure does not change over time. Instationary waves, however, have moving locations of minimal amplitude over time. For a time-dependent view of the pressure distribution in instationary fields, we animate the oscillating field based on the complex eigenmodes:

$$p(x, \phi) = re(x) \cdot cos(\phi) + im(x) \cdot sin(\phi),\ \phi \in [0, 2\pi] \tag{4.8}$$

For a phase shift of

$$\Delta\phi = cos^{-1}(im(x)/A(x)) = sin^{-1}(re(x)/A(x)),$$

this equation takes the form of a sine wave

$$p(x, \phi) = A(x) \cdot sin(\phi + \Delta\phi)$$

with a maximum of $|p(x, .)| = A(x)$, as visualized by a stationary view of the field. Resulting visualizations for different ϕ are shown in Figure 4.8, showing how amplitude minima evolve in the moving wave of the oscillating system for a frequency of 169 Hz and walls with high reflection coefficients. For the given example, the valley surface structures perform a rotation along the vertical axis. Corners, where amplitudes are generally high are untouched by the valley surfaces, an observation that verifies our visualization technique. The evolution of valley structures over time provides means to analyze wave propagation and

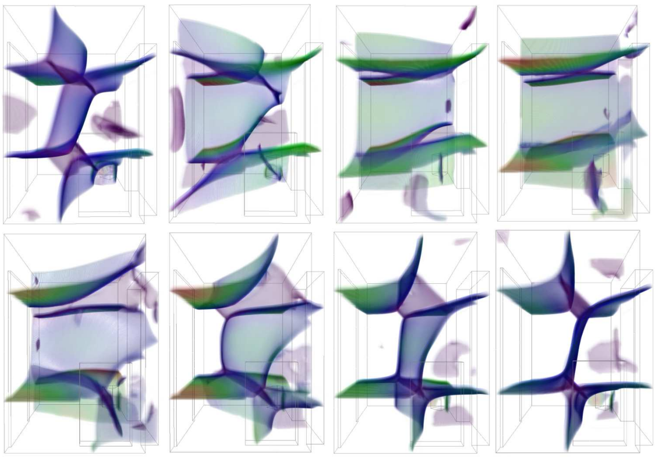


Figure 4.8.: Sequence of 8 non-manifold valley surfaces ($\Delta\phi = \frac{\pi}{8}, \pi \in [\frac{\pi}{8}, \pi]$), depicting node structures in an animated wave field. Color coding shows curvature magnitude. Pictures are volume-renderings of voxelized surface geometry.

sound cancellation for chosen frequencies. For a standing wave, no change in the valley can be observed, as its nodes are stationary complexes.

In order to give a spatio-temporal insight into what happens in a simulated room, nodes at different time steps can be accumulated as shown in Figure 4.9. To allow for volume-renderings and to facilitate different operations on this surface set for multi-field visualization purposes, we perform mesh voxelization on all individual surfaces, see Appendix B. While resulting set of voxels may be rendered using volume-rendering approaches and suitable transfer functions, more interesting results are achieved by the application of boolean operations on these voxel sets. Intersection operators allow the definition of multi-field features by highlighting minima that persist over time or selective slicing-plane based visualizations. The obtained visualizations show that regions far away from surface positions maintain high amplitudes over the whole oscillation cycle, while certain inner regions maintain a low amplitude throughout the whole cycle. Our three-dimensional depiction of these minimal structures allows the detailed inspection of inner wave structures, while reducing the amount of visual clutter and data that is prevalent in direct scalar- or gradient field visualization. Analysis of low amplitude regions over time allow optimization of room layout to avoid the appearance of undesired amplitude minima at listener positions. This is an advantage of time-varying valley surface extraction that is hard to achieve using conventional sound visualization techniques.

The stationary view of the same field is depicted in Figure 4.10, where extraction was performed directly on the amplitude values of the complex pressure field, thus giving an impression of where maximal amplitudes over time are minimal. In a standing wave, a stationary minimum surface of zero amplitude corresponds to

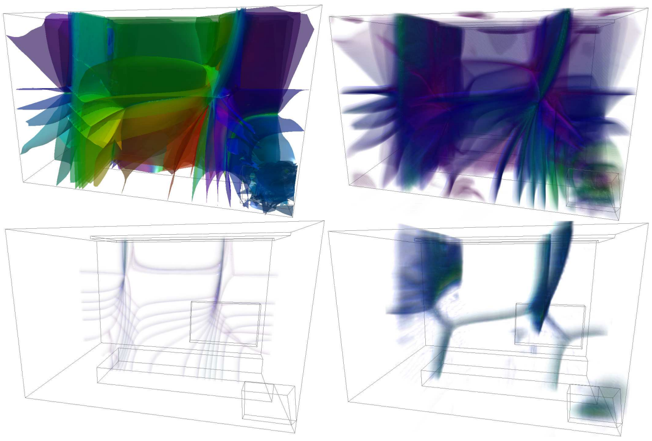


Figure 4.9.: Accumulated surfaces for 169 Hz with high reflection coefficients rendered as transparent surfaces. Valley surfaces are colored by time-step. Distinct regions of non-zero amplitude are found near corners of the simulated room. The resulting surface meshes can be rendered directly or be voxelized for volume rendering and boolean operations. Bottom row shows slicing plane and persistent minima obtained from surface voxelization.

wave-nodes. From an application side, this conveys important information about noise cancellation and wave topology. To give an impression of point distribution and mesh adaptivity, we render surface points as circles scaled according to the neighborhood radius used during seeding. Circles are colored according to size. Figure 4.11 displays the same situation for 137 Hz. We map different values to color to further emphasize properties of the valley surfaces. A manifold valley surface extraction in the l-shaped room example for 85.8 Hz is given in Figure 4.12 and compared to standard direct field visualization methods. It can clearly be seen that isovalue based methods do not capture all minima, as scalar values of minima can vary over the range of the data set.

Our results prove that analysis of simulated acoustics can benefit greatly from the use of valley surface visualization. The spatio-temporal visualization technique, for example, allows a target driven optimization of room acoustics if applied to different frequency bands. In acoustic design, localizing regions with low sound amplitude in both moving and stationary waves is essential for auditorium layout. With the corresponding simulation techniques, minima extraction may serve acoustic engineering in larger environment such as concert halls medium frequency ranges as well. The combined display of multiple fields in spatio-temporal visualization and in phase-based color coding further aids the understanding of field properties.

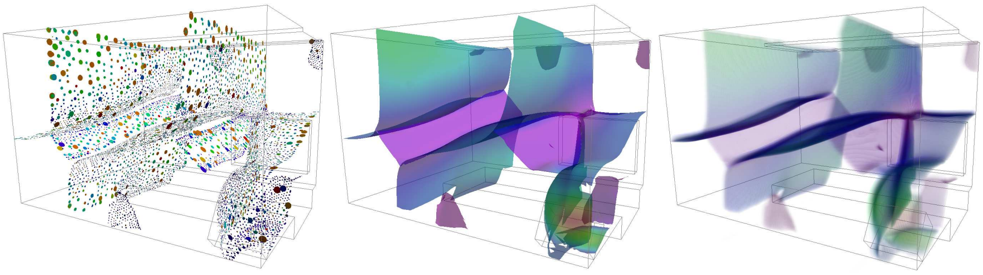


Figure 4.10.: Non-manifold mesh for a stationary view at 169 Hz. A non-manifold surface of minimal amplitude separates anti-nodes of the pressure field. Visualization techniques include point glyphs (left), direct mesh rendering (middle) and volume rendering of voxelized geometry (right). Color is mapped to curvature magnitude.

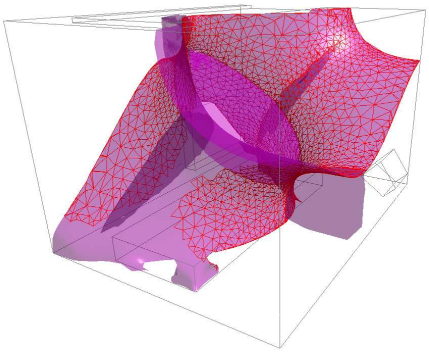


Figure 4.11.: Non-manifold mesh for a stationary view at 137 Hz. Part of the final surface triangulation with curvature dependent triangle sizes is shown, indicating a merging of triangulation fronts. Color mapping of amplitude values shows that the amplitude on most parts of the valley surface is virtually 0.

4.5.2. Evaluation

Figure 4.13 shows a comparison of a common grid-based extraction technique to our approach. The grid-based result is obtained from a modified version of the Marching Cubes based surface extraction with eigenvector orientation according to [KTW06] that operated on a tetrahedral mesh directly. Compared to our approach, it does not extract branching regions as seen in Figure 4.18 (b) and is not curvature dependent. For better comparison, we used the same curvature approximation technique in both cases, namely quadratic Moving Least Squares fitting. However, computation times of our approach are usually both higher due to repeated field evaluation during point convergence and depend directly on the complexity of the field, as it has an immediate influence on surface point numbers. Additionally, performance is dependent on the chosen scattered data

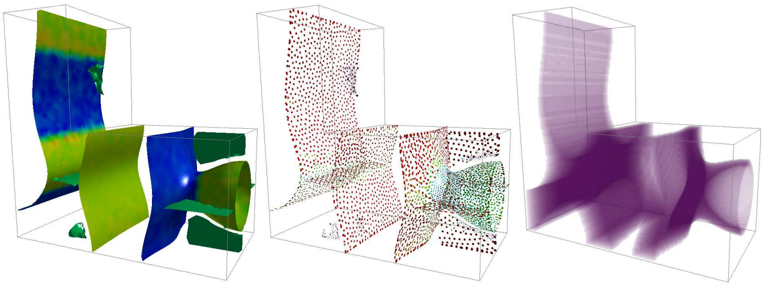


Figure 4.12.: Stationary disconnected valley surfaces for 85.8 Hz. Our region-growing algorithm extracted approximately 10000 points, giving an impression of three-dimensional wave nodes. Mesh coloring allows identification of regions with different phase values. Color on seeding ellipses represents average axis length and indicates refined mesh structures around highly curved areas. Conventional visualization methods such as thresholded volume rendering of amplitudes around 0 (right) make it hard to capture and analyze all minima.

approximation method and kernel size. We note that our technique recovers all true valley surface parts, while the grid-based approach creates false surface segments, by connecting independent valley points if different valleys cross the same cell of the mesh and is especially prone to noise in scalar field values. The creates additional surface parts have to be considered noise and need to be removed in a post-processing step. This difference becomes more obvious in noisy data sets as illustrated in Figure 4.14.

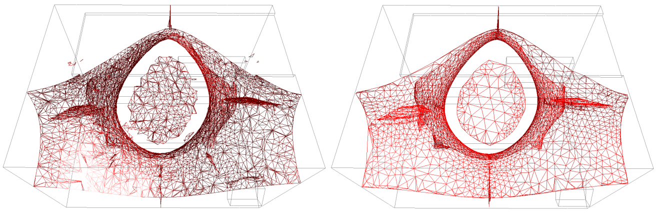


Figure 4.13.: Result of grid-based valley surface extraction with orientation criterion from [KTW06] and our approach (right). Our approach shows accurate boundary representation and uniform curvature dependent particle distributions, leading to a well-conditioned final triangulation.

Changing the influence of curvature on edge sizes during seeding facilitates easy control of mesh resolution prior to extraction as shown in Figure 4.16. Low resolution meshes show rough edges (see Figure 4.18 (a)) and may incorrectly merge different valley surface structures in case of unbounded edge lengths as seen in Figure 4.17 for a synthetic sinusoidal data set, while offering a way to quickly compute and identify valley structures. This fact is utilized for interactive surface

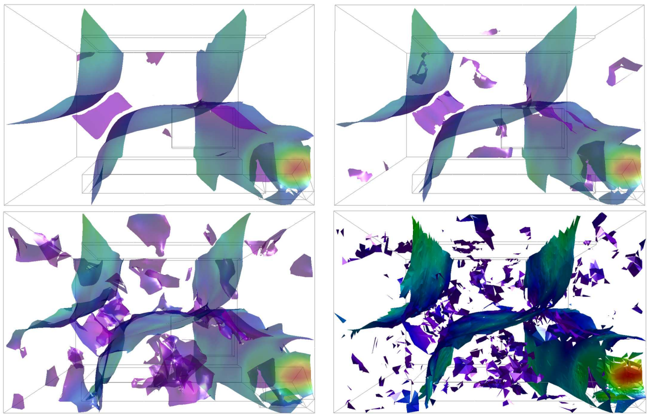


Figure 4.14.: Original valley surface and valley surface cores resulting from the addition of (5% and 10%) synthetic noise. Geometry of surface parts with low curvature magnitude indicated by color mapping is affected more strongly than parts with high curvature magnitude. A comparison to marching-cubes based surface extraction in the highly noisy case reveals differences in small scale noise caused by false positives and negatives during linear edge interpolation.

creation and refinement as described in Section 4.3.1.7. Figure 4.15 demonstrates, how interactive growing of low-res surfaces and subsequent refinement may be used for selective data exploration. The localized nature of these operations allows interactive update of the valley surface.

Tables 4.1 and 4.2 present computation times for a selection of mesh accuracy settings. The shown computation times were measured on a 2.16 Ghz Intel Core 2 Duo with 4 GB memory for the manifold extraction scheme. Tracing of accurate slope lines in the data sets increased computation times by a factor of approximately 5 due to repeated field evaluation during line integration. We observe that lod settings have a noticeable impact on mesh extraction times and mesh quality, whereas increasing the convergence parameter ϵ decreases the number of iterations performed during Newton iteration and therefore computation times but does not have a noticeable impact on mesh properties, as long as the value of ϵ is sufficiently close to 0. To compare the performance of the standard Newton iteration procedure to our proposed analytical minimum finding method, we have conducted the same computations with both methods (see Table 4.2). As expected, the analytical method profits from polynomial approximation and increases speed of convergence. To evaluate accuracy of the final surface, we compare surface mesh normals with ridge normal as computed by a difference quotient approximation of (4.3). While the resulting average normal deviations as listed in Table 4.3 do not provide information about the correctness of point locations of the extracted surface (this is guaranteed by choice of ϵ), it gives an

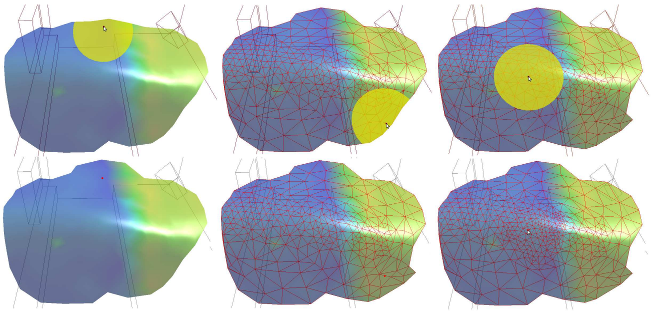


Figure 4.15.: Our surface growing approach allows selective surface construction and refinement. This facilitates interactive and progressive surface construction. Left two columns show user guided surface growing, right column shows interactive surface refinement of a surface color coded with phase. In the given example, the user is provided with a view-aligned cylindrical selection tool to mark regions for refinement or growing. Visual highlighting of selected regions is performed by OpenGL stencil-buffering.

Data set	high res	medium res	low res
L-shape	112s 4780 pts	66.8s 3040 pts	56.4s 1760 pts
137Hz	48s 4270 pts	15.9s 1310 pts	12.2s 880 pts

Table 4.1.: Extraction times and point numbers for different levels of detail using $\epsilon = 1e^{-8}$ and constrained Newton iteration for the examples shown in Figures 4.11 and 4.16.

insight into correct surface meshing and connectivity. The obtained values confirm that the extracted meshes are highly parallel to the true ridge surfaces. In cases of noisy data sets, surface quality is reduced due to less accurate curvature and ridge normal computations.

4.6. Summary and Discussion

We have introduced a new method for the mesh-free extraction and triangulation of valley and ridge surfaces that is able to create curvature adaptive non-manifold structures. For this purpose, we have analyzed the behavior of derived eigenvector and gradient fields, which allows extraction of maximal convexity ridges. The presented results show an improvement over the state of the art, as they allow accurate extraction of such features in scalar fields with arbitrary neighborhood structure. Moreover, the expressiveness of our approach in practical applications was pointed out in frequency-dependent complex valued pressure

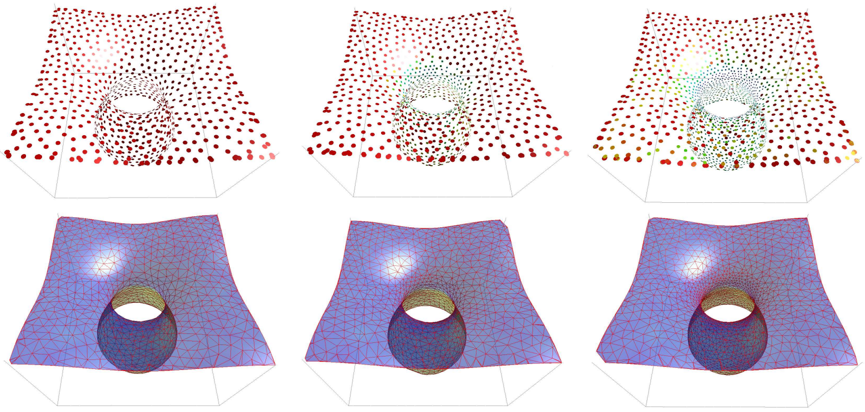


Figure 4.16.: Various versions of the same valley surface structure are obtained by varying the lod parameter. From left to right: low, medium and high resolution. Neighborhood icons colored by average axis length and surface triangulation show a significant decrease in point number and density around highly curved areas, as lower levels of details are chosen.

Data set	high res	medium res	low res
L-shape 137Hz	19s 2000 pts 25s 2900 pts	12.5s 1300 pts 13.6s 1700 pts	12.1s 1000 pts 5.9s 700 pts
L-shape 137Hz	31s 2000 pts 27s 2900 pts	20s 1300 pts 15s 1700 pts	16s 1000 pts 6.5s 700 pts

Table 4.2.: Extraction times and point numbers for the examples shown in Figures 4.11 and 4.16 for different levels of detail using $\epsilon = 2e^{-4}$ with quadratic minimum finding (top rows) and constrained Newton iteration (bottom rows).

fields from acoustic simulations. The extraction of node structures in these fields has proven to convey important and insightful information in both stationary and periodic time-varying cases. Amplitude and phase fields are conveyed by a combination of feature extraction and color coding. Furthermore, we show how the visualization of a set of periodic scalar fields by valley surface operations can help in understanding the time-varying behavior of the system. These multi-field visualization methods are based on components of the complex scalar field and are able to show features in amplitude and phase fields and can provide a spatio-temporal display of the time-varying system. For the visualization of multiple-fields in time-varying sound simulations, voxelization techniques help to perform binary operations on sets of single-field features and are able to yield occlusion-free multi-field visualizations.

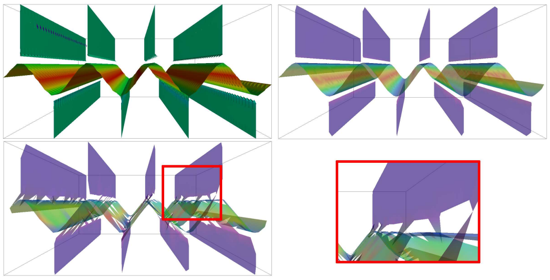


Figure 4.17.: Top row: Standard cell-based surface extraction and extraction with our method and appropriate level-of detail. For an appropriate choice of level-of-detail, valley surface meshes created by mesh-free and cell-based surface extraction are identical in noise free datasets. Bottom row: For inappropriate choice of level-of-detail, neighboring particles converge to different valley surfaces, resulting in jagged and ill-conditioned surface meshes. Coloring corresponds to curvature magnitude.

Data set	high res	(+ 5% noise)	low res	(+ 5% noise)
169Hz	0.121°	0.176°	0.123°	0.177°
137Hz	0.048°	0.14°	0.067°	0.142°

Table 4.3.: Accuracy of surface approximation reflected as average deviation between mesh normal and ridge normal for points on the surface mesh ($\epsilon = 1e^{-6}$). As expected noise and reduction of surface resolution have a negative influence on surface accuracy.

One topic of future research is the automatic detection of bifurcations, i.e. events where the wave structure changes when sliding through the time or frequency range. Another interesting question is the possible parallelization and accurate porting of mesh-less algorithms to the GPU.

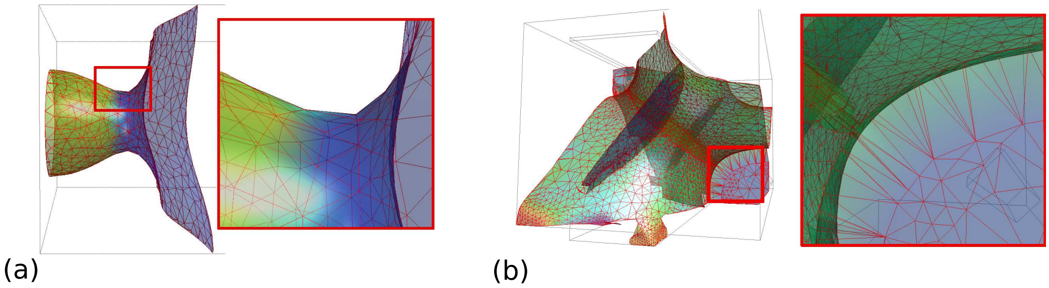


Figure 4.18.: (a) As the resolution of surface extraction is decreased, curved areas start showing edge artifacts due to the lack of curvature adaptivity and increased edge lengths. Phase color-mapping shows a clear transition between regions on the surface corresponding to different phase values. (b) Stationary mesh for 137 Hz with triangulated non-manifold regions color coded by phase. A zoom into a relevant region shows mesh elements created by the ribbon-triangulation procedure.

5. Strain in Flow Fields

The impact of tensorial measures and strain on flow analysis is often underestimated in the field of flow visualization. However, the increasing popularity of Finite Time Lyapunov Exponent maps in flow visualization emphasizes the importance of such strain measures, which is confirmed by a variety of research results that have been published on efficient extraction and visualization of FTLE maps and corresponding LCS [GGTH07, GWT$^+$08] in the last years.

These techniques build upon the fact that mixing processes form a part of flow simulations that are of direct significance to practical applications. The quality of mixing performed in a flow field is an abstract measure that is hard to capture by a single feature definition. While local properties of the flow such as divergence are fast to compute, they do not give an insight into the overall behavior of the field. Strain measures, however, may be combined to represent a more global view of the mixing process.

In the following sections, we present a new method for the efficient and accurate computation of flow induced strain and show its use in grid-less time-varying vector fields. This method examines deformations of infinitesimal particles by integration of strain tensors along individual integral lines of flow fields rather than relying on flow maps created by the advection of multiple neighboring particles. The created visualization techniques for time-varying integral lines do not only convey strain magnitude, but strain directions and anisotropy as well. Therefore, expressiveness of the resulting strain tensors includes the notion of FTLE maps and allows conclusions about convergence and divergence in flow fields.

While FTLE-like measures have been applied to a number of application fields for flow simulation such as aerodynamics and industrial mixing, we present how this flow induced strain analysis can help examine geophysical flow data. The evolution of strain and development of material anisotropy in mantle flow fields convey important information about geophysical properties of underlying geometry. We compute time-varying strain vector fields that build the foundation for a number of feature extraction and visualization techniques. The proposed field segmentation, clustering, histograms, and improved multi-volume visualization techniques facilitate a simpler and more intuitive understanding of strain in such flow fields, than provided by previous methods such as 2D line plots and slicing. We present applications of our approach to both an artificial time varying flow data set and a real world example of stationary flow in a subduction zone and discuss the challenges of processing these geophysical data sets as well as the insights gained.

In the following sections we first introduce the concept of strain advection along individual integral lines and present numerical challenges before introducing a concrete application where we use the resulting concepts and definitions for segmentation and analysis of geophysical mantle flow data.

5.1. Strain Advection

For the computation of FTLE maps [GGTH07], a set of neighboring particles is traced through the flow field for a finite amount of time. The gradient of the resulting flow map conveys important information about the convergence and divergence of these particle trajectories [Hal01]. We propose techniques for the accurate computation of similar continuum deformations by accumulation of strain tensors along individual integral lines, as illustrated in Figure 5.1. In the following sections we make use of the notion that second-order square tensors are able to represent strain information in the form of volume deformations as stated in Chapter 3, and detail the computation of localized strain tensors along individual integral lines.

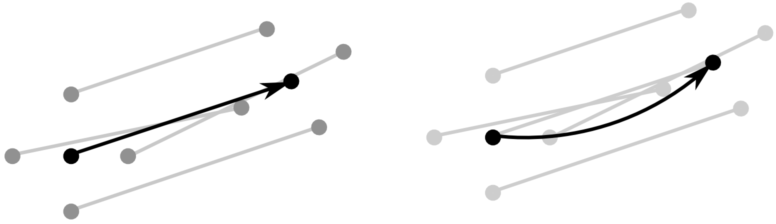


Figure 5.1.: Flow map computation for FTLE generation is based on particle set advection (left). For small particle neighborhoods, a comparable notion of deformation can be obtained by strain accumulation along a single particle trajectory (right).

5.1.1. Computation

In order to simplify notational and algorithmic complexity, we restrict the description of strain advection in the following to individual particle traces, i.e. strain along a stream- or pathline as investigated in the context of Eulerian flow fields in [AD85] or for stationary vector-fields in [Obe08], before presenting novel generalizations to sets of particle traces in the form of fully adaptive streaklines.

5.1.1.1. Strain Accumulation

Given a sequence of discrete positions $[x(t_0), x(t_1), ...]$ along a particle trace, the deformation tensor D defining the mapping from an initially spherical particle neighborhood at $x(t_0)$ to a deformed shape at $x(t) = x(t_{i+1})$ is computed by accumulation of strain information along the particle trace:

$$D_{x_0}(t) = \left(\prod_{j=0}^{i} exp(S(x(t_j)) \cdot (t_{j+1} - t_j)) \right) \cdot D_{x_0}(0) \tag{5.1}$$

with an initially isotropic shape or neighborhood model $D_{x_0}(0) = I$. The relative displacement tensor of the field at $x(t_j)$ is $S(x(t_j))$. The change in shape over an interval $[t_j, t_{j+1}]$ along the trace $[x(t_j), x(t_{j+1})]$ is captured by the matrix exponential $exp(S(x(t_j) \cdot (t_{j+1} - t_j)))$, as defined for a general matrix A by

$$exp(A \cdot \Delta t) = \sum_{i=0}^{\infty} \frac{1}{i!} A^i \cdot \Delta t^i \tag{5.2}$$

corresponding to the solution of the concrete linear ODE

$$\frac{dD_{x_0}(t)}{dt} = S \cdot D_{x_0}(t) \tag{5.3}$$

with $D_{x_0}(t_0) = I$. This mathematical relation is depicted in Figure 5.2.

In the following, we briefly show how the flow Jacobian is used as a linearization of this notion along streamlines. As the Jacobian describes relative displacement with respect to the canonical directions, it keeps all information necessary to observe rotating or shearing deformations.

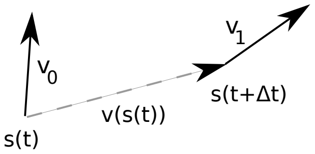


Figure 5.2.: Illustration of transformation of neighborhood vector for Euler-integrated integral line.

Given a differentiable velocity field $v : \mathbb{R}^n \to \mathbb{R}^n$ and a streamline $s : I \subseteq \mathbb{R} \to \mathbb{R}^n$, the flow Jacobian $J \in \mathbb{R}^{n \times n}$ on s describes transformation of neighborhoods along s. A vector $v_0 \in \mathbb{R}^n$ at $s(t) \in \mathbb{R}^n$ is transformed into $v_1 \in \mathbb{R}^n$ at $s(t + \Delta t) \in \mathbb{R}^n$ by the mapping:

$$v_1 = v_0 + (J(s(t)) \cdot v_0) \cdot \Delta t \tag{5.4}$$

For non-adaptive, Euler-integrated streamlines $\Delta t = 1$ holds for two subsequent points on the streamline, since each step of integration advances the streamline by $v(s(.))$, see Figure 5.2. This equation describes a first order approximation or linearization of the flow field at $s(t)$. To extend this vector mapping to a infinitesimal neighborhood around positions on s, we make use of tensor notation.

Representation of a number of vectors $w_i \in \mathbb{R}^n$ defined in the neighborhood of a point as a tensor with columns corresponding to w_i

$$D_{n \times n} = \begin{pmatrix} w_1 & \dots & w_n \end{pmatrix} \tag{5.5}$$

allows simultaneous transformation of multiple vectors and application of (5.4) to a local neighborhood:

$$\begin{aligned}
& D_1 && = D_0 + (J \cdot D_0) \cdot \Delta t \\
\Leftrightarrow\quad & D_1 && = (I + J \cdot \Delta t) \cdot D_0 \\
\Leftrightarrow\quad & D_{n+1} && = \prod_{i=0}^{n} (I + J_i \cdot \Delta t) \cdot D_0 \\
\Leftrightarrow\quad & D_{n+1} && = \prod_{i=0}^{n} \left(\sum_{j=0}^{1} \frac{1}{j!} J_i^j \cdot \Delta t^j \right) \cdot D_0
\end{aligned}$$

As expected, this equation contains a linearization of (5.2). The deformed volume at step n is therefore defined by the initial tensor and an accumulation of all transformations prior to step n. Consequently, strain accumulation along n segments of an integral line requires multiplication of n square matrices. It is important to note that the use of the Jacobian, i.e. the matrix of first order spatial derivatives, for this form of strain computation in time-varying fields requires $\Delta t << 1$ and low turbulence with respect to time in order to avoid approximation errors. Otherwise, derivation with respect to time is required.

5.1.1.2. Multiplication Scheme

After line integration and construction of the sequence of displacement tensors $M_i := (I + \Delta t \, \nabla v)$, they need to be accumulated to describe a univariate deformation along the integral line. We present a product length based matrix multiplication scheme to both reduce the resulting numerical errors and the space needed for matrix storage. A simple example of the general multiplication idea is shown in Figure 5.3. Compared to the canonical approach, the multiplication graph shown has a multiplication depth of $O(\log n)$ rather than $O(n)$, therefore reducing the accumulated multiplication error. As prior tensor data is constantly updated by matrix-product information results during the multiplication process, storage usage is optimized.

For a given sequence of n matrices, we construct an array of size n of sorted sets S_i of quadruplets. Such a quadruplet holds the following information:

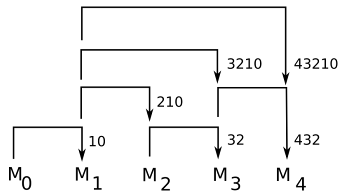


Figure 5.3.: Matrix multiplication scheme showing direction of multiplication, storage target and indices of accumulated tensors.

i	position in matrix sequence
start	lowest index of matrix in product
end	highest index of matrix in product
length	length of product

The initial distribution of quadruplet sets in the array for the given example is: $[\{(0,0,0,1)\},\{(1,1,1,1)\},\{(2,2,2,1)\},\{(3,3,3,1)\},...]$. Sets in the array are sorted by the length parameters of their members. The multiplication scheme works as shown in the following pseudo code:

```
while $\exists$ quadruplet k, k.start $\neq 0$ do
   for each quadruplet i do
      $j$ = first quadruplet of $S_{i.end+1}$
      if isMinSum(i.length + j.length) then
         $M_{j.i} = M_{j.i} \cdot M_{i.i}$
         Remove quadruplet j
         Insert l = (j.i, i.start, j.end, j.length + i.length) into $S_{i.start}$
      end if
   end for
end while
```

This scheme guarantees in every step that the pair with lowest total product length is multiplied and therefore reduces the depth of the multiplication tree to a minimum. While this leads to an increased number of matrix multiplications, compared to the straightforward approach, it avoids excessive accumulation of multiplication errors. Constant complexity due to mapping and queuing of quadruplet sets ensures that the computational overhead produced by the quadruplet data structure stays on a minimum.
Example of the scheme for a sequence of four tensors:

$\{(0,0,0,1)\},\{(1,1,1,1)\},\{(2,2,2,1)\},\{(3,3,3,1)\}$
$\{(0,0,0,1),(1,0,1,2)\},\{\},\{(2,2,2,1)\},\{(3,3,3,1)\}$
$\{(0,0,0,1),(1,0,1,2)\},\{\},\{(2,2,2,1),(3,2,3,2)\},\{\}$

$\{(0,0,0,1),(1,0,1,2),(2,0,2,3)\},\{\},\{\},\{(3,2,3,2)\},\{\}$
$\{(0,0,0,1),(1,0,1,2),(2,0,2,3),(3,0,3,4)\},\{\},\{\},\{\},\{\}$

This scheme is suitable for long particle traces as well as large sets of comparatively short particle traces, as created by adaptive line integration.

5.1.2. Strain Along Streaklines

Streaklines are connected sequences of particles released in a time-dependent flow field. Rather than representing the trace of an individual particle, they consist of a set of particles that follow different pathlines, as illustrated in Figure 5.4.

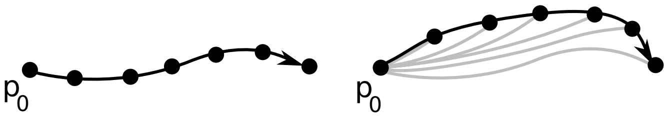


Figure 5.4.: For identical seed points p_0, pathlines (left) and streaklines (right) may vary substantially in the same flow field. Particle-traces for individual particles on the streakline are pathlines shown in gray.

While the mathematical foundation of strain deformation stays the same as with pathlines, the scheme of matrix-accumulation changes in adaptive streaklines. The sequence of tensors needed to define a particle deformation is in practice obtained by following a discrete particle through time and accumulating its Jacobians. As accuracy requires the generation of adaptive streaklines, leading to the insertion and removal of particles during line advection, not all particle-traces have the same length.

Particles which are created at the streakline seeding position start with an initial tensor as used in pathline strain accumulation. Tensor traces of deleted particles simply end, if the particle is deleted. Special care has to be taken while processing traces of particles, that were inserted during adaptive streakline generation. As initializing these particle traces with an identity matrix leads to inconsistent deformation information, a pre-deformed initial state has to be approximated from the deformation data present in the streakline. For this matter, we interpolate this initial state from tensor data available at neighboring particles:

$$T_0^r = (1-w)T_p^{r-l_1} + wT_q^{r+l_2}$$

where r is the index of the newly inserted particle. The particles $r-l_1$ and $r+l_2$ are the nearest left and right neighbors, that have not been inserted in this time step (i.e: $p,q>0$). T_0 is calculated by euclidean distance based linear interpolation of the two accumulated tensors. Componentwise interpolation of tensors is the most basic way of constructing intermediate tensor representations. If a consistent interpolation of eigenvector directions is desired, intermediate tensors can be constructed by more complex interpolation schemes [HSNHH10]. The tensor accumulation scheme is restricted to one time step in the stationary case and is therefore only required to evaluate one matrix sequence. In addition to the increased need of memory, a streakline occupies multiple time steps and produces a large amount of comparatively short tensor sequences for recently seeded particles.

5.2. Related Work

Work related to the scope of this chapter can be found in two different areas. The first set of related work originates from the field of visualization and is centered on visualization of vector and tensor fields and is in parts related to strain analysis. The second field is related to geophysics, where certain basic techniques are utilized for visual analysis of flow data. Even though the focus of papers from the latter set is mostly not on expressive visualization techniques of geophysical properties, but on geophysical analysis, we list them to provide the means of a comparison to common visualization techniques in this application area.

5.2.1. Visualization

Local flow properties such as velocity or divergence can be mapped to the radius of integral flow lines to produce *streamtubes* [USM96]. Other line-based deformation approaches visualize local magnitudes such as angle of rotation to ribbon like structures known as *streamribbons* [USM96]. A method that uses deformations of spheres to display local data along streamlines was presented in 1994 by Brill et al. [BHD$^+$94]. However, these mapping methods are usually applied to local information of the field and do not incorporate univariate functions such as accumulated deformation along an integral line. We show how this can be achieved with glyphs in stationary and time-dependent datasets. Since the work of Haller [Hal01] strain magnitude is widely regarded as suitable quantity to represent hyperbolicity in flow fields. These regions of extremal strain are commonly visualized as extrema in the FTLE field [GGTH07]. In contrast to our work, these visualizations do however not convey strain direction and are computed on sets of trajectories rather than on single integral flow lines. In the last years, strain segmentation has received increased attention in vector field visualization by the definition of various Eulerian and Lagrangian strain measures, which have been used to segment flow fields into strain and vortex regions based on the magnitude of these components [SWTH07]. Related research in tensor field visualization and *Diffusion Tensor Imaging* in particular [ZMB$^+$03, STS07, RJF$^+$09], uses anisotropy clustering based on tensor invariants with varying distance measures. For a state-of-the-art report on volume visualization we refer to [EHK$^+$04].

5.2.2. Geophysics

In geophysics, analysis of *Lattice Preferred Orientation* or grain orientation [ZK95] obtained from mantle flow simulations of subduction zones [JB10, WSMG06] is commonly performed by manual inspection of generic visualizations resulting from techniques such as slicing and arrow plot visualization [JB10] of instantaneous *Infinite Strain Axis* directions [JB10, KR02]. While these visualization methods are generally available and easy to implement, they suffer from loss of dimension and lack a homogeneous field impression. The need of manual parameter adjustment such as slicing plane orientation and position favors overlooking of features during visual inspection.

A first attempt at computing and interpreting grain orientations from time-varying flow simulations was done in [MvKK02] by the integration of strain along particle tracers. The resulting two-dimensional strain orientation field is visualized using line plots and temperature based coloring and serves as basis for further manual analysis and segmentation. Work with focus on visualization of geophysical flow with analysis of heat and critical point development is given by [EYD02]. A natural two-dimensional watershed-like visualization of grain orientations is given by optical micrographs with polarized light [ZK95] and is the result of photographing physically sliced material samples in lab experiments.

5.3. Strain Field Analysis

In the following, we introduce the presented technique to the visualization community by performing flow induced strain computation to construct a dense strain field in geophysical flow and present techniques for efficient flow analysis.

5.3.1. Motivation

Geophysics suffers from a lack of specific visualization techniques and requires the definition and development of alternative and tailor-made visualization methods. In geophysics, movement of our planet's inner structure is approximated by flow simulation of large sections of the Earth's mantle with high viscosity at varying temperatures. Not only is viscosity highly different from what is usual in other domains of flow simulation, but the same holds for time-scales. To be able to observe significant movements, time-scales of up to several thousand years have to be considered. As the point of interest shifts away from classic flow field illustration such as integral structure and vortex core extraction towards visualization of more general continuum mechanics, applicability of generic flow visualization techniques is rather limited. In the concrete application of geophysics, we are concerned with upper mantle flow in zones, where tectonic plates sink into lower regions of the Earth's mantle when converging towards each other (see Figure 5.5a). These subduction zones are of high importance to geologists and subject to intensive seismic analysis. While flow simulation models used in geophysics commonly provide users with a multitude of information on flow direction, temperature, viscosity and stress tensors, the obtained stress and strain values play a major role in the visual analysis of mantle properties, as they convey important information about seismic anisotropy and material alignment.

Our work aims at improving three-dimensional analysis of strain orientation in subduction zones by integrating a combination of feature extraction and field processing techniques with novel methods of strain segmentation and visualization. As a consequence, our work contributes both to the application field of geophysics by providing an intuitive and interactive visualization for flow induced strain analysis as well as to the visualization community by introducing novel methods for non-directed strain axis field segmentation, cluster visualization and selective multi-volume visualization of scalar fields.

5.3.2. Strain in Geophysical Flow Data

Subduction zones are seismically active regions of the upper mantle, where different tectonic plates meet. Over time-scales of millions of years rocks within the Earth deform viscously and flow like a fluid. Convection as the main force causing movement of tectonic plates is caused by heat emission of the core, radioactivity, and cooling effects present in subduction zones.

One of the main challenges of modern geophysics that is related to the Earth's interior is the inability to directly measure mantle flow direction and magnitude. The availability of such flow characteristics in response to subduction slab movement can provide important information about rheology of mantle rocks, composition of the mantle, and interplay between mantle flow and properties of tectonic plates [Bil08].

5.3.2.1. Background

In geophysics, the analysis of mantle flow pattern and preferred material alignment are closely related, as one observation can be used to approximate properties of the other.

In contrast to mantle flow directions, the preferred material alignment directions can be obtained directly from seismic measurements. Geophysical measurements of seismic wave propagation in the Earth's mantle can be used to infer the alignment of crystals of seismically anisotropic material and approximate direction of mantle flow patterns. As polarized seismic waves arrive at sensors at different times depending on the bulk alignment in the mantle, observed differences in the propagation speed allow computation of the fastest seismic axis in regions of the mantle. Geographical plots of these fast-axes serve as approximation of mantle flow directions [RS94]. In general, such seismic anisotropy is caused by deformation occurring in the upper mantle and causes the creation of preferred orientations in the crystal lattice.

Direct measurement of mantle flow direction is generally not possible and requires the use of numerical simulations instead. Numerical simulations of mantle flow and the analysis of accompanying material orientation patterns allow geophysicists to predict seismic wave propagation properties caused by earthquakes or artificial sources. Consequently, the relationship between deformation, mantle flow, and material alignment is of central interest in geophysics. This relationship depends on a number of other factors as pointed out in [KJKS08]. When neglecting microscopic influence on material orientation such as recrystallization, these directions of preferred alignment may be approximated by deformations caused by macroscopic external forces such as strain induced by the flow field [WSMG06]. As significant shearing or anisotropy is required to cause detectable material alignment, the magnitude of stretching is an important quantity as well.

Conventional visualization methods used for mantle flow and material alignment analysis in geophysics have limited previous studies, as they generally do not allow semantic linking of seismic lattice preferred orientation with numerical simulations of mantle flow in a three-dimensional setting. Consequently, most studies have made simplifications and focused on two-dimensional map slices or cross sections of the stretching directions (see Figure 5.6), ignoring inherently three-dimensional effects, or interpreting only large-scale patterns [CBS07, JB10, MvKK02].

During data set analysis, geophysicists have to identify regions with different strain orientation as well as regions with high anisotropy manually to make estimations about predicted seismic wave propagation. The inevitable requirement of manual plane alignment as well as visual ambiguities and clutter resulting from common visualization methods based on slicing complicates data set analysis. We aim at developing methods to assist and support geophysicists in their analysis of seismic anisotropy and mantle flow. Consequently, the proposed visualization methods presented in the next sections mainly rely on the data provided by the velocity vector field and the accompanying stress tensor data.

5.3.2.2. Data Format

Mantle flow simulation data in this work is given on an irregularly spaced curvilinear grid in a geographic coordinate system and represents mantle flow in a small section of Earth as seen in Figure 5.5a. For computational reasons, we transfer this geometric data in a pre-processing step from locally rotated geographic coordinate systems to a global Cartesian coordinate system according to

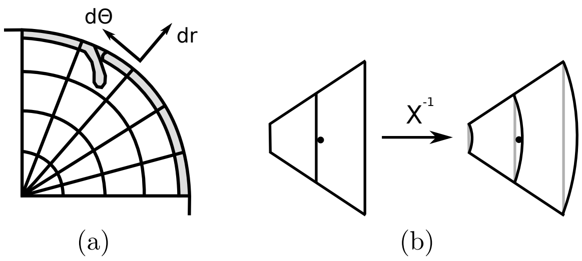


Figure 5.5.: (a) Two-dimensional illustration of possible data set locations in geographic coordinates. One section of the data set contains a subduction zone in the upper mantle.
(b) Point transformation is performed before cell-wise interpolation. This avoids incorrect cell identification and allows interpolation in geographic coordinate space.

standard coordination transformation rules. The neighborhood relation between data points is not affected by these mappings. Cells of the data set are identified by indices (i, j, k) with irregular spacings in longitude, latitude and radius direction. To guarantee correct cell identification and avoid errors that would occur when evaluating field values in the now distorted neighborhood grid, positions during field evaluation in $\mathbb{R}^3$ are mapped back into the curvilinear grid. The correct surrounding cell of the evaluation point is subsequently identified by binary search in the grid intervals of longitude, latitude and radius as illustrated in Figure 5.5b. Cell based interpolation is performed in local geographic space.

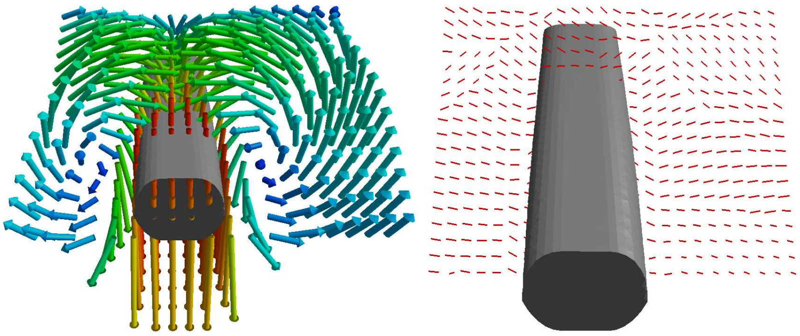


Figure 5.6.: Currently used slicing techniques in geophysics are able to show flow direction and magnitude by color coding (hue space) (left) and grain orientation (right).

5.3.3. Strain Field Computation

The general notion of a strain field as used in this work is that of a vector field whose orientation corresponds to the direction of main deformation of Lagrangian particles. To compute this deformation, we release isotropic particles at regions of inflow and deform them along their path according to strain in the flow field as introduced in a previous section. The strain direction is then computed from the resulting anisotropic shapes. These strain axes approximate preferred material alignment in geophysics caused by macroscopic external forces.

Simulations of the geodynamic model commonly return flow, temperature, viscosity and symmetric stress data. The stress-strain relationship of non-Newtonian fluids is non-linear, and strain at different stress levels exhibits varying behavior such as viscous or plastic deformation. If the model of stress-strain relationship is known, the strain rate tensor field can be accurately obtained from the input stress field and given viscosity, which lie around 10^{20} $Pa \cdot s$ for rock deformation. In current geophysical simulations, a viscous flow model is assumed.

In a more general case, the infinitesimal strain rate tensor may be computed by decomposing the velocity gradient of the flow field into a rotational antisymmetric and a symmetric part. For a three-dimensional vector field $v : \mathbb{R}^3 \to \mathbb{R}^3$, the velocity gradient may be approximated by finite differences and decomposed into the strain tensor and rotation tensor as defined in Chapter 3.

Once the data set is mapped to Cartesian coordinates, we perform strain analysis with a desired level of detail according to the following steps:

A uniform three-dimensional Cartesian grid with a user specified voxel resolution is imposed on every time step of the data set. In time-varying data sets pathlines are integrated such that every voxel in 4D is traversed at least once. Particles traveling along these integral lines are subject to strain forces and thus change from a spherical to a deformed shape. The main axes of these strain ellipsoids define an orientation field that approximates material directions at arbitrary points in time of the flow simulation. We subsequently segment the resulting strain axes field using different techniques to highlight distinct strain regions. Details on these segmentation steps are given in later sections.

5.3.3.1. Uniform Discretization

Geological data sets usually contain a large number of data points and field types to capture a meaningful section of the Earth's mantle. As our time-varying strain mapping method requires that every cell of the final strain field grid is traversed by at least one pathline, we let the user choose a resolution for this discretization to reduce computational complexity and support interactivity. Thus, we assume in the succeeding sections that the data set is superimposed by a Cartesian grid of $l \times m \times n$ voxels v_i per time step t. Resolution of the final strain orientation field is bound to the resolution of this discretization step, i.e. strain orientations are given per voxel by evaluation of the strain direction of the particle closest to the voxel center in space and time.

5.3.3.2. Pathline Integration

Strain orientation at a given point p in space and time is computed by accumulating strain information along a particle path that ends in p. As our method produces strain orientation values per voxel similar to other image based visualization techniques like FTLE, it has to be guaranteed that every four-dimensional voxel v_i^t is traversed by at least one such particle path. To satisfy this requirement, one particle path, i.e. one pathline, is started at the centroid of every voxel v_i^0 and integrated with an adaptive 4th-order Runge-Kutta integrator [TGE97] until it leaves the data set or reaches the last time step. During traversal voxels v_i^t are marked with the line and position index of the pathline, if a four-dimensional floating point rasterization [Bre65] of the pathline crosses the voxel.
New pathlines are started at all unmarked v_i^t and integrated in forward and backward direction. After this stage of the algorithm, every voxel is traversed by at least one pathline. Furthermore, every voxel knows the index of the traversing pathline as well as the closest preceding discrete position on that line as illustrated in Figure 5.7.

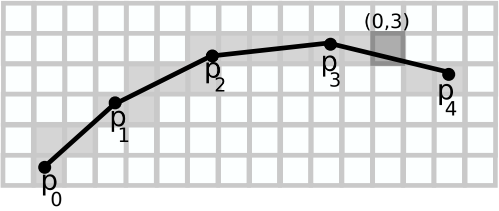


Figure 5.7.: Illustration of two-dimensional line rasterization. The highlighted cell knows line (0) and position index (3).

5.3.3.3. Orientation Field Computation

Let $p_x^t : \mathbb{R} \to \mathbb{R}^3$ denote the pathline with starting position $p_x^t(t) = x$. Under the assumptions that a pathline $p_{x_0}^{t_0}$ crosses a given position x^t at time t and that initial particle shapes are isotropic, strain orientation at x^t is defined as the direction of the major axis of the deformed particle at $p_{x_0}^{t_0}(t) = x^t$.

Again, length and direction of the major axis of the deformed neighborhood $D_{x_0}^{t_0}(t)$ at x^t are given by the square root of the maximum eigenvalue and the corresponding eigenvector of $D_{x_0}^{t_0}(t)^T \cdot D_{x_0}^{t_0}(t)$.

Based on these definitions and the information collected during particle tracing, it is feasible to compute the desired strain axis orientation field for arbitrary time steps. Values of the desired strain axis orientation field are computed at the center of every voxel v_i^t by evaluating (5.1) at the closest point on a pathline that crosses this voxel and interpolating it in a nearest neighbor fashion. The direction of the major axis represents our strain axis orientation at v_i^t, magnitude of the field corresponds to the largest eigenvalue of $D^T \cdot D$.

5.3.3.4. Stationary Strain Fields

Unlike the Lagrangian strain analysis presented in the preceding sections, geophysicists often simplify grain orientation computation by analyzing instantaneous strain distributions. Further reasons to work with stationary flow fields are the large time-scales as well as missing availability of time-varying data sets. The concept of *infinite strain axis* analysis [KR02] facilitates local strain axis computation that successfully approximates a complicated Lagrangian strain advection in many real world situations. Thus, the instantaneous strain orientation field is calculated as a local approximation of the asymptotic major axis of the strain ellipsoid after an infinite amount of constant strain.
The methods described later in this chapter are suitable for analysis of the resulting infinite strain orientation fields.

5.3.4. Strain Field Segmentation

As the main direction of strain corresponds to expected lattice preferred orientation, strain segmentation in the context of this work focuses on this direction only [ZTW06], rather than on other tensor properties [LGALW09]. The set of strain orientations in a given time step defines a non-directed vector field $g : \mathbb{R}^3 \to \mathbb{R}^3$. This strain field contains valuable information about approximated material alignment and strain states. Seismic analysis is concerned with identifying lattice preferred orientations and deducing appropriate mantle movement. In the following we present two criteria for strain segmentation to aid analysis of important grain orientation properties. Positions with non-unique strain orientations in the case of isotropic deformation are in the following treated as separate group, i.e. they exhibit maximal dissimilarity to other strain orientations.

5.3.4.1. Orientation Segmentation

To allow analysis of regions of grain orientation discontinuities, we capture rapid changes in grain orientation or by computing the Frobenius norm of the Jacobian of the normalized gradient field $g^* = \frac{g}{\|g\|}$:

$$f_o(p) = \sqrt{\sum_{i=0}^{3} \sum_{j=0}^{3} \left(\frac{\partial g_i^*(p)}{\partial x_j} \right)^2} \tag{5.6}$$

Due to the normalization of g, $f_o(x, y, z)$ represents a measure of angular deviation of the strain axis field g in a neighborhood of p. Thus, maxima of f_o

describe regions with a sudden change in strain orientation. We approximate the Jacobian by difference quotients with locally oriented strain axis directions.

The voxel grid together with values of f_o creates a three-dimensional scalar valued image that can be used as basis for domain segmentation. One such image based segmentation technique is known as *watershed segmentation* and its capability to extract locally maximal structures fits our need of feature extraction [RM00]. To filter small scale noise, we apply an anisotropic diffusion filter [PM90] to the image contents that preserves prevailing edges while smoothing small scale noise and perform a watershed segmentation using ITK [ITK05]. As common in ridge-based segmentation techniques, parameters of this watershed segmentation allow user-guided definition of minimal ridge height.

The resulting three-dimensional image contains full segmentation information for a given ridge strength and could be rendered using conventional direct volume rendering. However, without further processing, these methods are highly non-interactive and fail to convey important boundary information. A solution to these issues is the extraction of individual geometric volumes from the watershed image and is presented in the next section.

5.3.4.2. Alignment Segmentation

A second important property of strain fields is the degree of alignment between strain and velocity direction. This is especially important in areas, where one of the two fields is used to derive information about the other, as is the case in seismic analysis of grain orientations. For this matter, we map g to a scalar field representing the angular difference between strain and velocity:

$$f_a(x, y, z) = acos\left(\frac{|g(x, y, z)^T v(x, y, z)|}{||g(x, y, z)|| ||v(x, y, z)||}\right) \tag{5.7}$$

Segmentation of the range of f_a induces a segmentation on the strain field itself. A segmentation of the strain field with respect to f_a is performed by applying thresholding to the range of f_a, thus subdividing the alignment interval $[0, \pi/2]$ into smaller segments. In contrast to FTLE visualization, where segmentation or highlighting of strain magnitude is desired, f_a can be used to separate flow parallel deformation from flow orthogonal regions of deformation. Additional visualization techniques as detailed in the next section allow incorporation of classic FTLE visualization techniques into segmented data sets. High values of f_a indicate a strong divergence between strain and flow direction and can serve as hint how to interpret measured grain orientation in corresponding regions of the data set.

5.4. Visualization

Visualization of a strain field requires fundamentally different techniques than the visualization of individual traces of deformed particles. We present methods for glyph based individual trace visualization and techniques that allow complete strain field visualization.

5.4.1. Trace Visualization

When inspecting single particle traces, concrete strain and shape properties are of central interest. For this reason, we use data provided by accumulated deformation tensors to visualize the shape of implicitly defined volumes along time-varying integral lines, as previously proposed for strain deformation in stationary fields [Obe08]. We specify an initial neighborhood shape that is deformed along particle paths by application of the linear mappings defined by (5.1). The initially isotropic neighborhood is chosen to be

$$M_0 = \begin{pmatrix} w_1 & w_2 & w_3 \end{pmatrix} = r \cdot I$$

with $r > 0 \in \mathbb{R}$, which implicitly defines a spherical glyph S as illustrated in Figure 5.8.

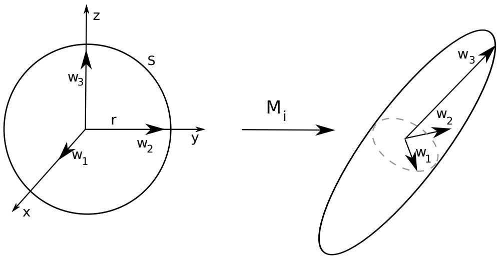


Figure 5.8.: Initially spherical neighborhood S is transformed into an ellipsoid.

Matrices M_i, $i \in \{0, ..., n\}$ in the particle trace correspond to a linear mapping of vectors $w_j \in \mathbb{R}^n$ on this sphere. While M_i is generally not symmetric, the ellipsoidal shaped resulting from a linear mapping of the sphere are point symmetric. The resulting ellipsoids can be rendered efficiently by linear transformation of precomputed sphere geometry. While these ellipsoids are simple, when compared to other tensor glyphs [Kin04], they convey most important tensor characteristics such as anisotropy and major, medium, and minor strain directions. Furthermore, more complex tensor glyphs tend to quickly lose characteristic shape features when viewed from a distance.

Besides visualizing strain directions, we wish to quantify the effects of strain on a given particle. For ellipsoidal volumes in incompressible flow, the deviation from an isotropic shape can be measured by computation of the quotient between axes lengths. As stated in Chapter 3, the Singular Value Decomposition allows computation of these axes for asymmetric tensors such as the ones generated during strain accumulation. Singular values of matrices $M_{n\times n}$ correspond to half-axes lengths of the implicitly defined ellipsoids.

Thus, the quotient between the smallest and the greatest non-zero singular value is a valid measure of maximal directional anisotropy. This quotient is directly mapped to hue in HSV space to show anisotropy magnitude, as seen in Figure 5.14.

5.4.1.1. Finite Integration Lengths

As ω is a direct indicator of the degree of deformation of a spherical volume, a value close to zero can be used to identify positions in the tensor sequence, where associated ellipsoids start to degenerate to flat or line-like shapes. Once such a position is found, subsequent matrix multiplications are redone with a reset initial tensor M_0, thus resizing any subsequent volumes. Whether one chooses to use this information for resizing is however application dependent, since even degenerate volumes might deform back to ellipsoidal objects, as long as the tensor data is valid, i.e. columns of the tensors are linearly independent. If a global finite integration constant is chosen, logarithms of the resulting strain magnitudes for a set of integral lines correspond to scaled FTLE values.

5.4.2. Strain Field Visualization

Glyph based visualization method are generally not suited for dense visualization of three-dimensional data. This insight and the fact that strain field analysis in the context of this work is concerned with macroscopic field segmentation requires the definition of alternative visualization techniques.

During strain field segmentation we identified several requirements for visualization of strain volumes. A visualization technique for strain analysis in mantle flow fields should be able to convey the following properties:

- show strain volume boundaries
- illustrate strain orientation and magnitude
- show relative flow directions

In three-dimensional data sets common challenges in the context of visualization are information overload, visual occlusion and clutter. In contrast to common strain visualization techniques that focus on glyph or hyper-streamline techniques [HJYW03] to visualize full tensor information, the techniques presented in this work combine visualization of major strain axis direction and field segmentation.

We propose visualization techniques to overcome these challenges, while focusing on depiction of mentioned properties of the data set.

5.4.2.1. Strain Lines

When neglecting field segmentation, the obtained strain field for one instance in time corresponds to a typical eigenvector field as obtained from tensor data such as DT-MRI and represents a general non-oriented vector field. Similar to streamline based visualization, this field type allows extraction of integral lines along strain directions. To visualize strain directions for a given time step of the simulation, we integrate a uniformly spaced set of lines L_i that are locally tangential to the direction of strain. These strain lines are then rasterized into a three-dimensional texture, with opacity and color at a position $p = (x, y, z)$ defined by

$$Rgba(p) = \begin{pmatrix} 2 \cdot f_a(p)/\pi \\ 0 \\ 1 - 2 \cdot f_a(p)/\pi \\ ||g(p)|| \cdot max_i(\omega(d(p, L_i))) \end{pmatrix} \tag{5.8}$$

where d returns the distance between a point and a given line and $\omega(d) = e^{-\frac{d^2}{r^2}}$ is a spherical kernel function with constant radius r. A volume rendering of the resulting three-dimensional strain texture produces pictures comparable to volume LIC [CL93], where strain magnitude/FTLE values are mapped to opacity and color to the degree of alignment between flow and strain direction.

5.4.2.2. Multi-Volume Visualization

The distinct regions in image space as obtained by the methods described in the previous section, are sets of voxels and as such represent geometric volumes that segment the present data set into separate sections. To visualize both strain orientations and boundary shapes of segmentation geometry, transparent rendering of these volumes is required. Correct rendering of segmentation geometry in the form of transparent boundary surfaces may be performed by methods such as depth-peeling [Eve01] or order dependent rendering. However, when combining this visualization with a volume rendering of internal strain orientation, one quickly runs into problems with respect to correct transparency rendering [RTF$^+$06]. As a consequence, we propose a multi-volume visualization technique related to the work proposed in [HBH03] for consistent and interactive rendering of both internal strain and segmentation geometry based on volume slicing.

A subvolume V_i defines a binary mapping on the data set, classifying individual voxel centers as interior or exterior, and keeps information about its bounding box extensions (d_x, d_y, d_z), position p_e in $\mathbb{R}^3$ and relative position p_v in discrete voxel space (see Figure 5.10a). We store values of this binary map $m_i : \mathbb{R}^3 \to \{0, 1\}$ as

three-dimensional mask in the form of alpha values of a RGBA texture, where entries of the RGB vector in the vicinity of volume boundaries represent geometric normals of the volume geometry, as seen in Figure 5.10a. While masks representing a disjoint decomposition of the field can be stored in a common texture, we decide to use separate mask textures to avoid invalid normal interpolations between neighboring volumes. Bounding-box attributes p_e, p_v, and d are stored as separate vectors. Trilinear interpolation on m_i yields a representative volume definition for an isovalue of 0.5.
Utilizing this data and the well-known Marching Cubes (MC) algorithm, we perform the following steps for slicing-based [EHK+04] selective multi-volume visualization:

1. Embed volume set into a virtual bounding box to determine sampling distance and consistent scalar values.
2. Determine view dependent distance values and propagate scalar values to bounding box corners for MC slicing (see Figure 5.10b).
3. Compute MC slice geometry t_j for all volume boxes and isovalue $iso = d_{max} - \Delta d$.
4. For every t_j: pass corresponding volume mask, position and triangulation data to shader.
5. Repeat from 3. with $iso = iso - \Delta d$

Slice geometry for a volume V_i obtained in step 3 is processed in a fragment shader as described in Algorithm 5.9 with corresponding mask and position data. Embedding the volume representations (e.g: bounding boxes) into a global context allows consistent propagation of scalar values in step 2. Line 5 of the shader algorithm ensures that only that part of the slicing geometry which lies within the corresponding subvolume is drawn, i.e. all geometry and volume information outside is clipped by an isovolume [WEE02] in the fragment shader. Isovalue stepping together with "z-less or equal" depth buffer testing guarantees view-dependent and consistent back to front rendering of volumes with correct transparency processing, see Figure 5.10c. To highlight region boundaries, volume boundaries are phong-shaded with pre-computed normals. We note that MC slicing can be based on an arbitrarily complex geometric representation of the volume to avoid execution of the shader on redundant fragments, rather than performing it on the complete individual volume bounding box. Making use of all available texture units can reduce the number of volume mask texture swapping, and further reduce computation times. This novel multi-volume visualization technique allows arbitrary selection and positioning of individual parts of the volumetric data set, has the advantage of preventing classic z-fighting issues when using pre-sorted triangulated geometry, and avoids expensive sorting or intersection computations, allowing highly interactive framerates.

Figure 5.9.: Pseudo code for pixel shader.

```
EXTERN tex3Dstrain, tex3Dmask, $p_v$, $p_e$, $d$
INTERN fragpos   //3d fragment position
$pos \leftarrow (fragpos - p_e)./d$;   //position in mask texture
$maskval \leftarrow RGBA(tex3Dmask, pos)$;
**if** $maskval.a > 0.5$ —— $outsideBB(pos)$ **then**
  discardFragment();
**else**
  **if** $maskval.a > 0.5 - \epsilon$ **then**
    phong(maskval.rgb);   //shade border of volume
  **else**
    $c \leftarrow RGBA(tex3Dstrain, (fragpos - p_e) + p_v)$;
    drawFragment(c);
  **end if**
**end if**
```

./ denotes component wise division

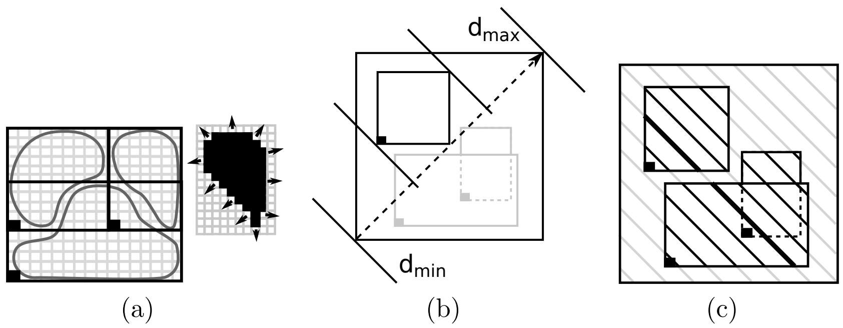
Interaction As the proposed multi-volume visualization method allows selective display and transformation of parts of the data set, we implement picking of single volumes. To realize color based picking in OpenGL, we extract geometric triangulations of volume boundaries as depicted in Figure 5.11. In contrast to the multi-volume visualization method presented in the previous section, this geometry is used for fast picking operations only and is not used for final visualization of the data set.

To provide a clearer look at volume boundaries, we give the user control over volume positioning by making use of the explosion metaphor, allowing volume positions to change according to a force driven concept.

Let $c_i^0 = c_i$ be the bounding-box center of strain volume $i \in \{1, \ldots, n\}$. Iterative evolution of volume centers is governed by:

$$c_i^{k+1} - c_i^k = h\frac{1}{\sum_j^n \omega(j,i)} \sum_j^n \omega(j,i) \frac{(c_i^k - c_j^k)}{||c_i^k - c_j^k||}$$

where $h > 0$ controls explosion step size and ω is a distance based weighting function. Visual distinction between different strain volumes is aided by color coding. For non-strongly interleaved volume sets this positioning concept provides better views at volume and boundary shapes. In addition to position control, color



(a) (b) (c)

Figure 5.10.: (a) Subvolumes in the data set define masks, positions in $\mathbb{R}^2$ and (static) positions in voxel space $p_v^1 = (1/16, 1/16)$, $p_v^2 = (1/16, 6/16)$, and $p_v^3 = (10/16, 6/16)$, as indicated by black corners. 3D texture mask stores normal information and binary relation, as shown for the upper right volume.
(b) View dependent scalar- and isovalue propagation for MC slicing.
(c) Changed positions in $\mathbb{R}^3$ still allow correct multi-volume visualization, when consistent slicing is applied. Extracted slice geometry for one isovalue is highlighted.

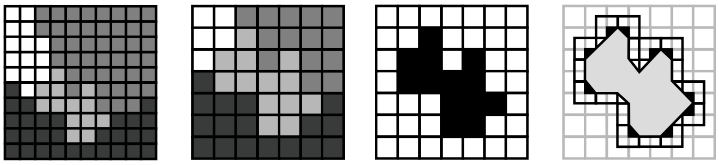


Figure 5.11.: Two-dimensional illustration of the steps of geometric volume extraction from a watershed image: 1. Region selection 2. Region masking 3. Isosurface extraction on the dual grid (resulting isocontour shown in gray).

based picking facilitates manual selection and toggling of volume display. These approaches allow a clear look at strain volume boundaries both by direct rendering of volume geometry, as well as positioning and selection of individual volumes. Additionally, transparency of volume boundaries is a free, user controlled parameter and interactive adjustment of volume transparency does not only give a clearer look at otherwise hidden volume boundaries due to the reduction of occlusion, but enables the user to have a look at interior properties of the volume by revealing strain lines.

5.4.2.3. Strain Histograms

While previously presented methods give a localized view of strain orientations, a more global view of these orientations and their segmentation is desired in strain analysis as well. An established display method for global segmentation analysis in the field of image processing are histograms. Histograms in image analysis perform interval clustering on the image range to visualize the relative frequency of certain color or intensity ranges. Fortunately, this form of binning is not limited to scalar fields, but may be generalized and adapted to higher dimensional data types such as vector fields. Three-dimensional spherical histograms have recently been used to illustrate global vector field behavior in time series data [GJL^{+}09]. We use a similar technique for spherical histogram construction in our non-directed strain axis field.

Sphere Parametrization Previous histogram methods use a uniform parametrization in spherical coordinate space to obtain bin intervals, leading to a high variation in bin domain sizes. Bin areas around the pole are degenerate and minimal, whereas areas along the equator are comparatively large (see Figure 5.12). This variation is opposed to the desire to use histograms as a consistent density measurement and visually cannot be fully compensated by scaling of bin heights. As a result of these observations, we prefer an almost equidistant parametrization of the sphere as given by icosahedron subdivision. The resulting sphere subdivision resembles a geodesic dome consisting of almost equilateral triangles, thus allowing uniform data binning of almost identical base area. Further advantages of this parametrization are consistent triangular bin shapes in contrast to mixed rectangular and triangular shapes as shown in Figure 5.12.

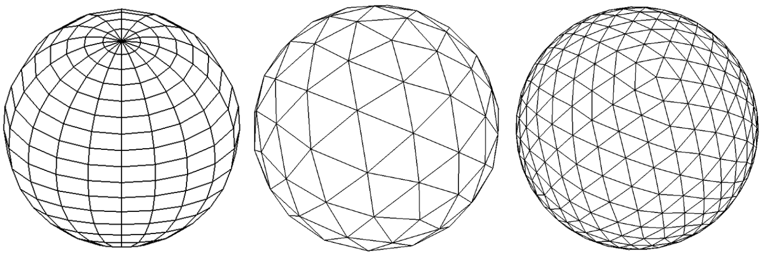


Figure 5.12.: Uniform parametrization (left) and two different levels of icosahedron subdivision.

Data Binning The orientation information of the strain axis field is reflected in the spherical histogram by mapping the height of right prisms to the number of orientation vectors pointing to its base triangle. This construction leads to a point-symmetric histogram in the case of non-directed vector fields. Volume membership information gathered during strain field segmentation is mapped to the histogram by the use of volume colors. , where each volume is assigned a distinct hue from HSV color-space with maximal saturation. During data binning, final color for a bin b is mixed according to strain volume membership. For this matter, we average all contributing color vectors of the base triangle in L$*$a$*$b$*$ space. The resulting color is transformed to HSV-space and saturation magnitude is divided by the number of volumes with significant contribution to the bin. This step ensures that visually unsaturated colors correspond to mixtures of multiple volumes, whereas saturated colors reflect dominant contributions of one volume. This guarantees that mixed colors do not accidentally correspond to original volume colors. These color finding steps are illustrated in Figure 5.13. We note that we limit initial volume colors to colors with high value components in HSV space to guarantee that a decrease in saturation leads to visually distinct colors.

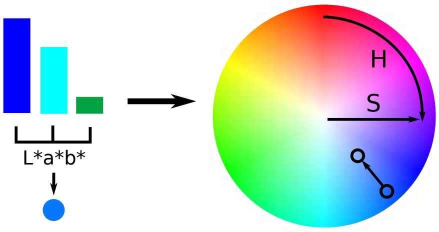


Figure 5.13.: Steps of bin coloring.

A histogram with a number of small distinctly colored regions with homogeneous color indicates that strain variation within single volumes is small.

Interaction The spherical histogram associated with the data set responds to user selections in volume visualization mode. Contribution of orientation vectors located in hidden volumes, i.e. volumes with toggled visibility, are removed from the histogram bins. Fast computation times during histogram updates are guaranteed by storing volume contributions for every individual bin and volume during initial histogram construction. Removal of certain volume contribution in bin height and color can thus be performed quickly without triggering a full recomputation of the histogram. Reduced histograms enable the user to quickly observe strain orientations in a more local and focused context.

5.5. Results and Application

We start by presenting results from the visualization of individual particle traces in general flow fields. If not stated otherwise, scalar valued quantities such as velocity or strain magnitude are mapped to hue in HSV space.

Figures 5.14-5.16 show practical results of tensor deformations in two- and three-dimensional cases. The visualizations indicate a mixing and disturbing motion near the flow obstacle. This is due to no-slip properties of the boundary surface. Specification of the starting points for stream- and streakline integration should either be done manually to analyze specific regions of the flow field, or can be done automatically in a uniform fashion to give an overall impression of the mixing properties of a certain flow field. In contrast to FTLE methods, strain ellipsoids allow the analysis of strain directions. For line or surface-like integral flow features, our method is able to to convey information about tangential and orthogonal strain. For example, Figure 5.16 shows prevalent strain in direction of the streakline, which is confirmed by streak surface refinement in the same direction as seen in Chapter 6. The resulting visualizations display data from multiple fields, namely the flow field and the derived strain tensor field. Over time (or space), the shape of deformed tensors tends to lose expressive power as it may degenerate to long or flat shapes. In such cases different points on the surface have low topological correspondence and strongly diverging trajectories. If this occurs a reset of the initial matrix of the following matrix sequence can produce a better visualization of the desired properties and resulting strain magnitudes are comparable to FTLE values.

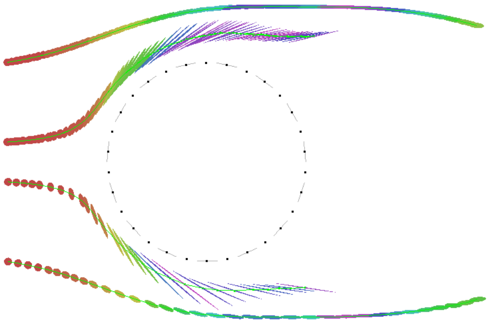


Figure 5.14.: Four 2D streaklines integrated past a spherical-obstacle. High resolution of lines gives a homogeneous impression, while low point resolution gives a good impression of general deformation properties.

This remainder of this section discusses both inherent visualization capabilities of the presented strain analysis method as well as visualization benefits. Due to the difference in strain field computation, we present results of stationary flow fields separately from the ones obtained in time-varying strain analysis.

5.5.1. Instantaneous Flow

The given instantaneous flow field covers a portion of a model of the southern Alaska subduction zone where the Pacific plate subducts beneath Alaska. Subduction slab geometry is approximated by temperature isosurfaces. This model has flow that is drawn in perpendicular to the top of the slab (poloidal flow) and flow around the edges of the slab (toroidal flow). An interesting feature is a region of slab-parallel (along the subducting plate surface in a horizontal plate) stretching surrounded by stretching that is caused by the poloidal flow and sinking of the slab. This region of slab-parallel stretching corresponds to the observations of seismic fast axis with similar alignment. This is an important

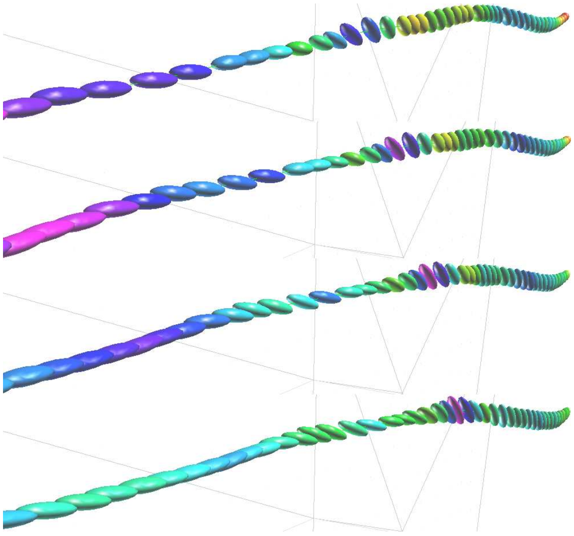


Figure 5.15.: 3D streakline with ellipsoids in a sequence of four consecutive time steps. Clinching, stretching as well as rotational behavior over time and space can be observed. Coloring indicates areas with heavily deformed volumes.

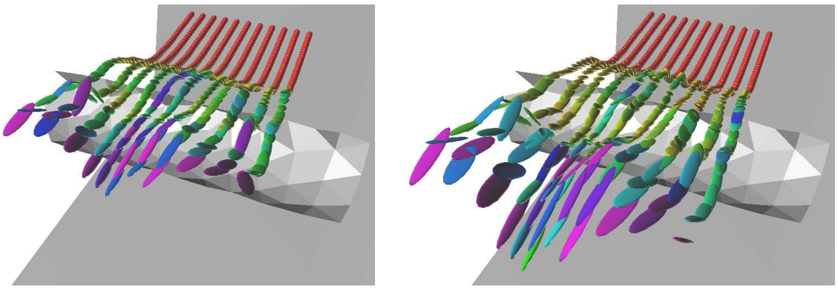


Figure 5.16.: Deformation in time-varying surface-like structures can give an insight into complex tangential and orthogonal strain behavior.

example of stretching and lattice preferred orientation that is perpendicular to the dominate flow direction of the mantle [JB10].

The data set consists of around 5.7 million points carrying velocity, temperature, viscosity and stress information. Furthermore, a pre-computed set of infinite strain axis direction, as described in Section 5.3.3.4, is provided. Data set extensions are $[23°, 29°]$ for co-latitude $[205°, 220°]$ for longitude and $[5971\ km, 6321\ km]$ for radius. Results of our segmentation method on a 100^3 voxel grid show expressive segmentation of the infinite strain axis field, identifying important features such as aforementioned slab parallel stretching (purple lateral volume in Figure 5.17). Figures 5.17 and 5.18 show the benefit of interactive picking and volume explosion. Further visual improvement is gained by transparent rendering of volume boundaries, providing a look at inner strain directions. We note that our volume visualization of the segmented data set with fully opaque volume boundaries visually corresponds to a triangulation based visualization. Figure 5.19 shows a comparison of different voxel resolutions. We conclude that low resolution f_o segmentation of strain orientation fields give a rough approximation of strain segmentation while generally producing more volumes than high resolution versions. Moreover, volume segmentations converge as the resolution increases towards the inherent data set resolution.

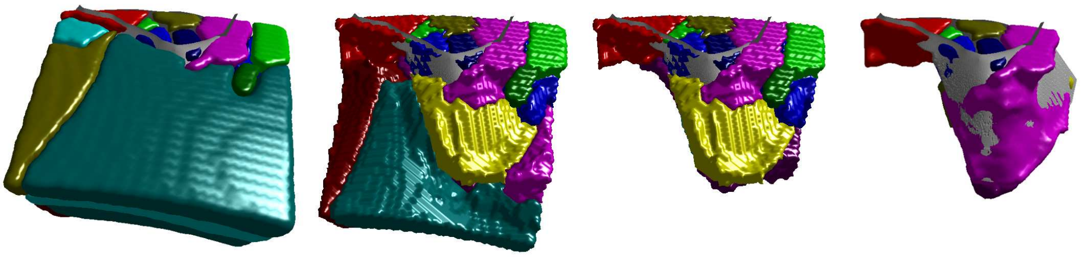


Figure 5.17.: Different f_o volumes in a 100^3 voxel grid. Volume masks of the first and last images were smoothed with a 3×3 Gaussian kernel. User selection of different volumes allows a clear look at inner segmentation, where strain segmentation coincides with slab geometry.

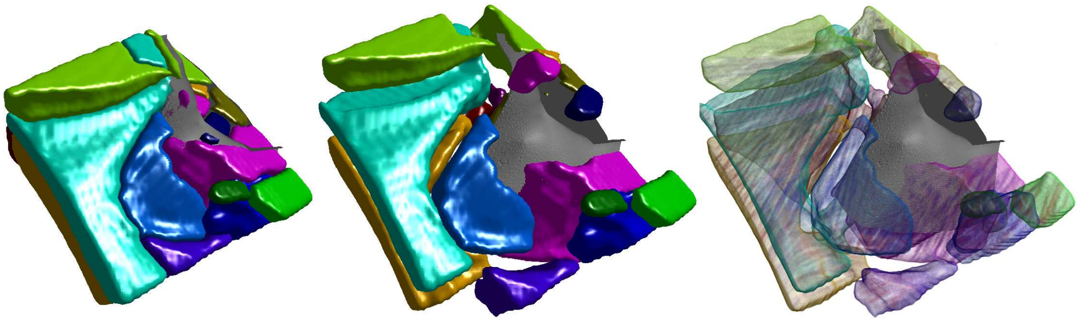


Figure 5.18.: Top view of Alaska data set. Mantle geometry shown in gray is modeled by a temperature isosurface. Explosion style rendering of f_o strain volumes provides a better look at volume boundaries. Additional transparent rendering reduces occlusion.

5.5.2. Time-Varying Flow

The second data set is a small region of a test model used to study the dynamic process of the detachment or break off of part of a sinking tectonic plate from the plate at the surface, as seen in Figure 5.20. In this test case, the sinking tectonic plate is comparatively young with an age of around 20-30 million years, which means that it is fairly thin and weak compared to older tectonic plates. Therefore

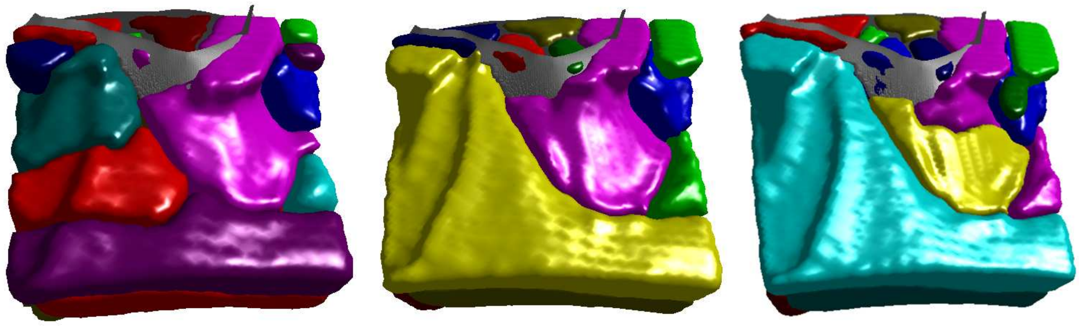


Figure 5.19.: Strain volume extraction for different grid resolutions (from left to right: 70^3,90^3,100^3). Lower resolutions yield different segmentations while reducing segmentation times. However, boundary shape is approximated well in all resolutions. For higher resolutions, f_o segmentation solutions converge.

the sinking plate rapidly sinks vertically. The flow of the mantle around the slab is again dominated by two components: poloidal flow and toroidal flow, as seen in Figures 5.20 and 5.21.

The time-varying data set provides 10 time steps of around 310000 points each. The total simulation covers around 2.4 million years with step sizes of an average of 270000 years.

Figure 5.21 shows a volume visualization of rasterized strain lines by reducing opacity of volume boundaries to 0. High strain magnitude is observed in immediate proximity of the slab. The stretching orientations and opacity encoding of magnitude show that the maximum stretching occurs where regions of poloidal and toroidal flow merge. Our visualization conveys convergence and divergence flow properties, as regions with dominant opacity correspond to Lagrangian Coherent Structures in FTLE fields. In regions with strong alignment between flow and strain (red) strain lines correspond to streamlines, whereas flow is orthogonal to strain in blue regions. This color coding can be used to identify relative flow directions. A visualization of two f_o subvolumes that does not make use of strain line opacity mapping is show in Figure 5.22. While strain orientations are clearly depicted in the final result, no information about magnitude is conveyed.

Figure 5.23 is a visualization of four selected f_o subvolumes of this data set. Homogeneous and consistent strain orientations can be seen on lateral parts of the slab path both in the volume display as well as in the histograms. Discontinuities in strain orientation develop mainly along paths of vortex cores. These prominent strain volumes are picked and visualized in Figure 5.24, where localized histogram information gives an insight into expected seismic anisotropy for different regions. Besides discontinuities along the sides of the slab, a horizontal discontinuity is created where flow is pulled down behind the sinking slab.

Results obtained by alignment segmentation are given in Figures 5.25 and 5.26. Strong disagreement in flow and strain alignment as extracted in Figure 5.25 is observable along outer regions of flow vortices and underneath the slab. Figure 5.26 shows 4 f_a segments and a selection of two volumes with weak and medium alignment. In contrast to f_o segmentation, histograms in alignment based segmentation show no clear segmentation in orientation space.

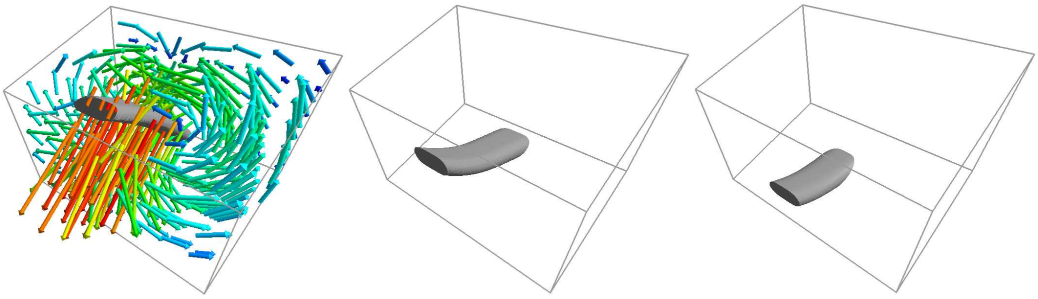


Figure 5.20.: Sinking slab in test data set. Velocity magnitude is highest along the path of the slab, while slow rotating flow is observed along its sides.

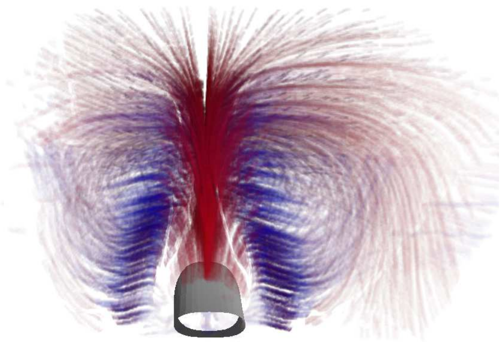


Figure 5.21.: Volume visualization of strain lines with fully transparent volume boundaries. Transparent discontinuities in the volume visualization are caused by small spaces between adjacent volumes.

From a performance point-of-view we note that for low resolution voxel grids ($< 50^3$) in the time-varying data set approximately 80% of computation time is spent on pathline integration. This percentage increases for higher voxel resolutions and more time-steps and can be observed in all FTLE-like flow analysis systems that utilize dense integral line generation.

Our results show that the presented extraction and visualization techniques are suitable for strain based analysis of time-varying and stationary flow. The examples analyzed in this work demonstrate that geophysical flow analysis benefits from these techniques by providing means of grain orientation discontinuity and

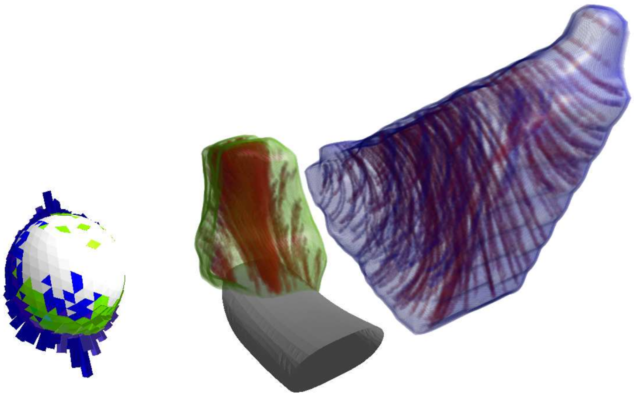


Figure 5.22.: Two subvolumes generated by f_o segmentation and according histogram with turned off strain line opacity mapping.

flow alignment visualization, thus allowing detailed conclusions about the quality and confidence into flow reconstruction from grain orientation and formation of separate material sections.

5.6. Summary and Discussion

We conclude that integration of flow induced strain along integral lines facilitates the analysis of mixing along particle trajectories. We presented accurate treatment and computation of particle deformations and demonstrated the use of ellipsoid based deformation visualization in time-varying data sets. The resulting deformation data can be interpreted as a localized version of FTLE, as was shown in a follow-up work by Kasten et al. [KPH+09]. Strain ellipsoids are an expressive tool for multi-field visualization if not only flow direction and particle trajectories, but strain field properties are to be visualized as well.

We have presented a method for automatic strain analysis in three-dimensional flow data and its application to the field of geophysics. Gridded pathline integration and strain accumulation in connection with mapping of the major strain axis field allows image based segmentation of the underlying orientation field. To analyze main strain directions in distinct strain volumes, we proposed the combination of transparent multi-volume visualization with picking and color mapped spherical histograms. This provides the user with the possibility to quickly identify the main strain directions and variance of a set of volumes.

Our work contributes to the geophysics community by providing an inherently three-dimensional view of the data and supporting automatic strain based analysis and segmentation. The methods developed in this work to extract and visualize regions with uniform stretching directions together with the magnitude of stretching provide means to identify regions with potentially strong material

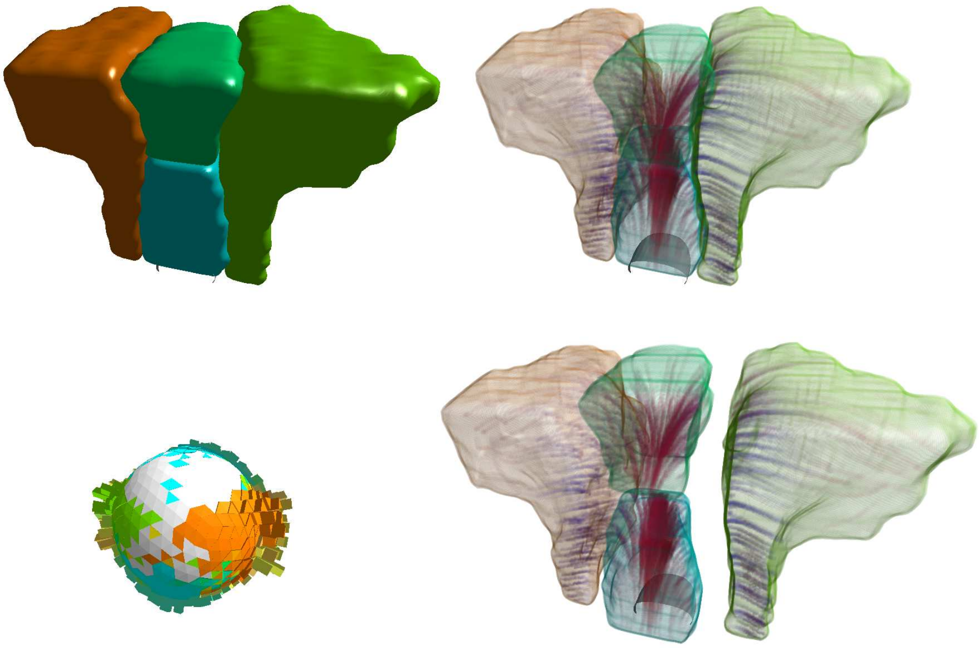


Figure 5.23.: Selection of prominent f_o feature regions. Volume rendering conveys the impression of phong shaded solid geometry as well as transparent volumes with interior strain lines. Explosion and volume selection allow an unobstructed view at volume boundaries. Further global and selective strain analysis is aided by spherical histogram visualization.

alignment. The developed visualization techniques are able to extract and show the geometry and locations of discontinuities and rapid changes in the derived strain field, the magnitude and orientation of the stretching, and the relative behavior of mantle flow within each region. This combined visualization is a major improvement over previous visualization techniques, as it allows linking of material alignment with flow directions as well as clear depiction of discontinuities. Identification of regions with uniform material orientations and visualization of magnitude and direction of strain can be used for the prediction of seismic fast axis orientations that are expected to be observed at the surface and can serve as basis to compare simulation data to seismic measurements and link material orientation to mantle flow direction. This information facilitates iterative comparison of numerical models and seismic observations and allows determination of flow patterns in the mantle.

The contributions of our work to the visualization community are the development of a robust and expressive visualization technique with focus on interior and boundary information representation for flow induced strain in both instantaneous and time-varying flow fields, created by novel strain extraction and seg-

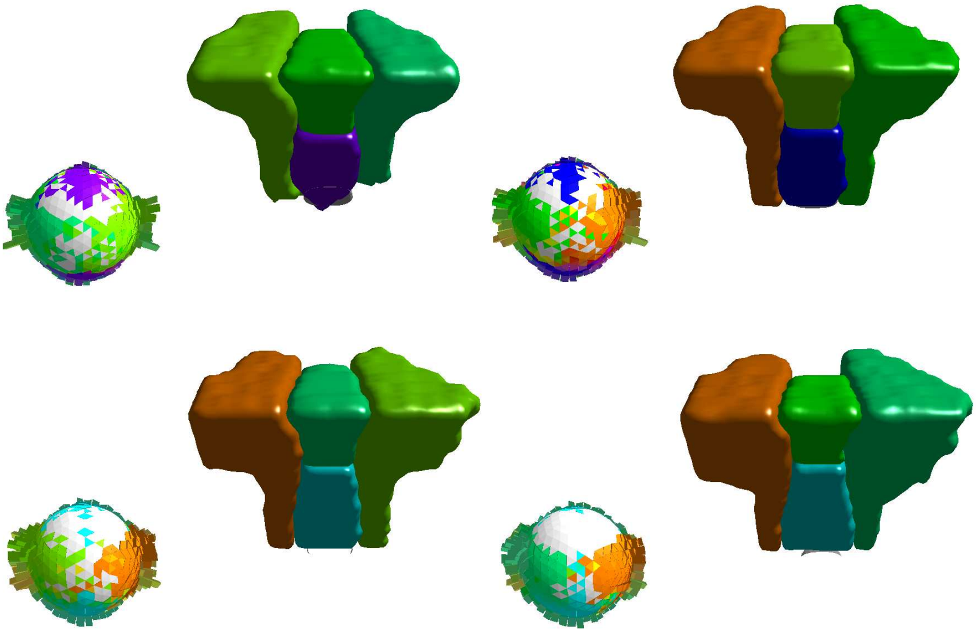


Figure 5.24.: Evolution of prominent f_o features and histograms over time. Histogram information conveys the fact that strain in the two lateral regions is orthogonal to strain directions in the center. Segmentation indicates that discontinuities in material alignment are formed along paths of the vortex cores and remain stable over a number of timesteps.

mentation methods and selective volume visualization techniques. The resulting visualization is able to convey information about two related fields: The strain direction and magnitude field and relative velocity information.

While we have demonstrated the applicability and expressiveness of strain analysis in geophysics, there are many other application areas that can benefit from a sophisticated strain field analysis. Besides extending and evaluating the applicability of our methods to other application areas, directions of future work in the field of geophysics may include incorporation of user defined or simulated seismic waves into the segmented field to directly illustrate the expected impact of volume boundaries and strain direction on wave propagation.

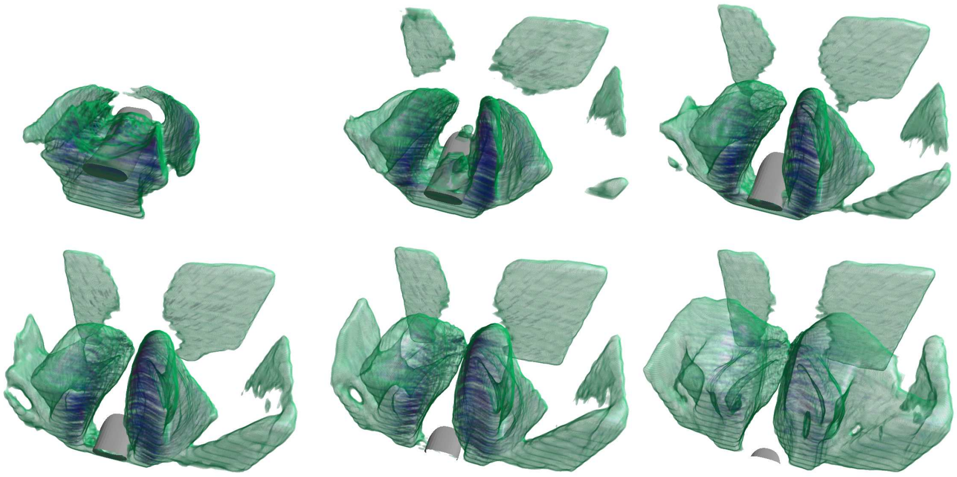


Figure 5.25.: Evolution of f_a region with maximal deviation between strain and flow direction. Maximal orthogonal strain is observed along frontal lateral parts of the slab and weak orthogonal strain in lateral parts.

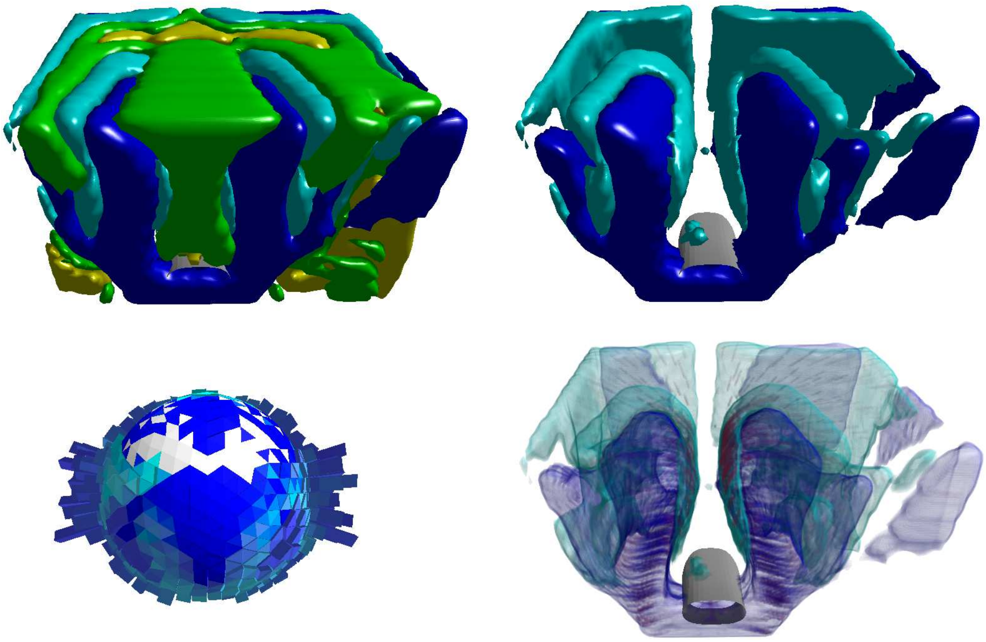


Figure 5.26.: Solid and transparent rendering of two f_a alignment segments in different time steps. Strain histogram and strain lines show non-homogeneous strain directions in all volumes. Strong alignment of flow and strain directions is observed along the path of slab movement, whereas disalignment prevails in lateral regions.

6. Unsteady Flow Segmentation

The concept of continuum deformation in the form of deformation analysis for infinitesimal particles can be transferred to a macroscopic view, where the behavior and spatio-temporal evolution of large particle sets is of interest. In unsteady flow, particle sets that are continuously released into the flow field describe streaklines and streak surfaces. Generation and rendering of integral curves and surfaces in simulated flow fields is a well-established technique in the area of vector field visualization, where the homogeneous visual properties of their geometry allow an in-depth analysis of the behavior of connected components of flow fields. Besides giving an impression of general flow behavior, tessellated integral features are suited for field segmentation.

In visualization (topological) segmentation of stationary flow fields is a well-covered topic, relying on the extraction of adaptive separatrices [OKHBH09]. In the following sections, we develop novel techniques for the segmentation of two- and three-dimensional time-varying flow fields by means of adaptive integral flow features. In comparison to curve and surface definitions where merely the trace of a specific single particle or curve is tracked, as is the case in path surfaces, the definition of streak lines and surfaces is capable of visualizing phenomena such as smoke and dye-advection, efficiently showing the movements of distinct, continuously seeded regions over time. In practice, they may be used to visualize material boundaries in mixing processes or, more general, a time-varying analogon of the stationary separatrix definition, leading to a concept related to stationary vector field topology. Therefore, streak lines and surfaces form the basis for (topological) area and volume segmentation in time-dependent vector fields.

In two-dimensional flow fields, integration of closed adaptive and boundary-integrated streak-lines at material interfaces facilitate the segmentation of the flow domain into streak areas for means of multi-field visualization. Multi-field visualization benefits from streak area extraction as it allows selective application of transfer functions.

In three-dimensional data sets, adaptive generation of streak surfaces poses additional challenges. While efficient methods to adaptively generate integral surfaces in stationary vector fields and some generalizations like adaptive time surfaces are state-of-the-art, there are no known methods for fully adaptive streak surface generation as presented in this chapter. An important application of surface feature extraction is the visualization of separation topology in three-dimensional data sets what, due to the absence of adaptive streak surfaces, has been limited to the stationary vector field case so far.

The major challenge of adaptive streak surface construction is the high complexity of mesh refinement arising from the new time dimension of the well-known advancing front definition introduced by Hultquist [Hul92]. In every time step of a time-varying velocity field, not only a front of a few streamlines or particle traces needs to be examined, but the whole surface has to be analyzed with respect to adaptivity measures. We solve the problem of adaptive streak surface integration with the help of MLS approximation for particle advection as well as surface estimation and refinement based on Delaunay meshes. Again, the presented test data sets are mesh-less and obtained from grid-less FPM. Delaunay meshing helps to approximate surface particle densities and provides a basic triangulation for visualization. The challenge of time coherent surface generation and rendering is overcome by surface particle integration, adaptive particle tracing, and look back methods.

In the following sections we establish a notion of the state-of-the-art in adaptive integral flow feature extraction, provide details about advection of individual particles before detailing refinement and segmentation techniques for two- and three-dimensional flow fields.

6.1. Related Work

Feature based topological segmentation of two-dimensional flow fields has been examined in [LL99] by Leeuw and Liere, who use a critical-point approach for segmentation purposes, what we try to avoid in this work. In [WTS+07], generalized streaklines are seeded at moving singularities. Adaptive placement of streaklines for texture-based flow visualization is introduced in [SMA00]. More recent work than the one presented in this chapter implements adaptive streaklines on the GPU [CPK09].

For three-dimensional fields, adaptive (stream-) surface generation in stationary flow fields was introduced by the work of Hultquist [Hul92] in 1992, whose well-known advancing-front concept has led, among other things, to the development of sophisticated methods for path surface integration in time-varying flow fields [GKT+08, STWE07]. The work by Krüger et al. [KKKW05] represents the most direct way of streak generation and visualization, namely by the use of large particle systems being influenced by the surrounding flow field. The absence of a triangulated mesh does however limit these point sets to discontinuous representations. Operations such as surface intersections are not possible without further effort. Funck et al. [vFWTS08], Cuntz et al. [CKRW08], and Weiskopf et al. [Wei04] introduced work on smoke surfaces and particle level set advection, focusing on the visualization of non-adaptive streak surface like structures. The first of these papers triangulates the particle system for visualization purposes and is therefore closer to our method, as far as visualization techniques are concerned.

Examples for MLS based surface approximation in the context of surface mesh reconstruction and refinement from point clouds are given by the work of Alexa et al. [ABCO^{+}01] and Mederos et al. [MVdF03].

Delaunay type mesh refinement of static point sets has been the topic of multiple papers such as the work of Chew et al. [Che93] and Chen et al. [CB97]. Both approaches use incremental mesh construction by either building a constrained Delaunay triangulation or by transforming common Delaunay algorithms to a new parametric space. Contrary to this work, the nature of our problem allows us to make use of an existing time-varying triangulation, thus eliminating the need for complete mesh reconstruction and facilitating the incorporation of multiple Delaunay triangulations into the particle insertion process. Parallel to the publication of the work described in the following sections, a similar approach to adaptive streak surface generation was proposed by Krishnan et al. [KGJ09]. Their approach however refrains from performing curvature dependent refinement and uses highly resolved time axes for surface refinement.

In the following sections we briefly present the used particle advection scheme and establish a notion of the definition and importance of adaptivity for integral flow feature generation. Furthermore, we give details about the integration of adaptive streaklines and transfer these concepts into three-dimensional space by proposing a novel adaptive streak surface generation technique. The results demonstrate the expressiveness of flow segmentation performed by these adaptive flow features.

6.2. Particle Advection

A requirement for the generation of artifact free streaklines and surfaces is the accurate advection of particles in unsteady flow fields. Since particle advection is a well-covered topic in unsteady flow visualization literature [KKKW05], we limit this section to a brief discussion of the techniques used in this work.

Based on the sampling frequency of the data set with respect to the time axis, a discrete time-dependent dataset can provide more or less visual coherency between adjacent time steps. In unsteady particle advection, particle data of time step $t-1$ has to be propagated to the next time step t and particle positions need to be advected to their respective new positions in t according to the velocity values obtained from vector field evaluation. To compensate possible artifacts in poorly time resolved data sets or data that has a high curvature in time, special care has to be taken during vector field approximation. While adaptive Runge-Kutta approximation schemes can determine the step size or order of integration k used during particle advection, the necessary interpolation between adjacent time steps that needs to be performed, if $k > 1$ often introduces artifacts if the degree of interpolation is too low, as in (6.1).

$$f(p, t-1+\frac{j}{k}) = \left(1-\frac{j}{k}\right) f(p, t-1) + \frac{j}{k} f(p, t) \tag{6.1}$$

where $f(p, t-1+\frac{j}{k})$ is the velocity of a particle with position p and integration order k during the j-th step of advection from $t-1$ to t. This scheme subdivides the time interval $[t-1, t]$ into k time intervals with linearly interpolated velocity fields. To reach a sufficient accuracy, we determine k for every particle individually by comparison of angular deviation between velocity vectors resulting from consecutive vector field evaluations. This approach has a similar effect on accuracy as adaptive integration of streamlines with Runge-Kutta or comparable methods in the stationary case. Higher order integration methods such as cubic Hermite interpolation, which take into account $f(p, t-2)$, $f(p, t-1)$, $f(p, t)$, and $f(p, t+1)$ for particle advection from $t-1$ to t double the number of required field evaluations and require more time-steps to be present in system memory, but tend to yield more accurate particle tracing.

6.3. Adaptivity

Our notion of streaklines and surfaces in the following does not include disconnected particle sets, but requires a continuous representation in the form of curve or surface geometry. The construction of such geometry and connectivity information facilitates and requires the discussion of adaptivity in integral flow features.

In a dataset with an equidistant timeline, n particles are generated at each time step at a seeding structure and subsequently advanced through time and space. Therefore, an ideal or *true* streakline or surface in a flow field is obtained for $n \to \infty$. However, an increase of particle numbers causes significant increase in computation times caused by the increased number of field evaluations that are necessary for particle advection. Furthermore, even high-resolution integral flow features can yield highly varying distributions of particles in regions with high divergence, which is often undesired for accuracy reasons.

Without the insertion of new particles during curve and surface evolution, the geometric representation of the integral feature diverges strongly from the true feature in regions of turbulent flow, as distances between neighboring streak-particles increase. Methods for adaptive streakline and surface refinement aim at resolving these issues by heuristic insertion of particles during advection. Furthermore, particle deletion in oversampled flow regions reduces computational complexity of particle set advection.

6.3.1. Insertion of Particles

Numerical integration of non-adaptive curves and surfaces reveals two problems that lead to inaccurate representation of the true feature:

1. *Segment Lengths:* Large distances between neighboring particles of a streakline or surface increase the risk of missing important flow features, as neighboring particles might pass on different sides of turbulent flow features. Such excessive element lengths on the geometric representation are often caused in situations, where particles traverse regions of strong divergence or flow regions with large strain magnitude orthogonal to the geometry tangent.
2. *Line Smoothness:* Distinct peaks in the geometric representation of the curve or surface represent regions, where particle traces behave differently from their neighbors. These regions are of high importance for the visual representation and integration accuracy.

Both situations indicate regions, where the flow field is not accurately sampled with particle pathlines. In both cases, additional particles have to be inserted at positions near the questioned regions in order to maintain a given particle resolution and guarantee a satisfyingly smooth representation of the flow feature.

6.3.2. Deletion of Particles

Insertion of particles in turbulent flow tends to increase particle number exponentially. For this reasons, the removal of particles needs to be incorporated into the feature generation process to reduce computational costs. Robust particle deletion has to obey the rules of conservativity. Therefore, particles should not be deleted if small perturbations in the resulting geometry lead to re-insertion of new particles, as this increases computational overhead and reduces overall accuracy.

6.4. Adaptive Streak Areas

We present techniques for adaptive streakline integration in two-dimensional flow fields and along boundary geometry, whose segmentation capabilities result in flow segments called *streak areas*. This work is motivated by the challenges posed by time-dependent vector fields to the problem of curve and surface extraction. We demonstrate, how the geometry of the resulting integral flow features is well suited for the segmentation of flow fields. In the two-dimensional case, traditional approaches to the extraction of separation geometry depend on separation line or entry-/exit point tracking. In the following sections, we present a technique to integrate adaptive streaklines along boundary geometry to avoid the tracking of separation geometry and incorporate the boundary into the integration process. We furthermore present a texture based visualization method to display time-varying streak areas.

6.4.1. Streakline Integration

As indicated by the streakline definition given in Chapter 3, streaklines are generated by releasing and advecting a stream of Lagrangian particles into a flow field. To find a balance between fast computation of low-resolution streaklines and accuracy of high-resolution curves as detailed in section 6.3, we propose the following refinement techniques:

6.4.1.1. Refinement

The adaptivity criteria introduced in Section 6.3 imply the insertion and deletion of particles at positions throughout the streakline.

Insertion of Particles We use the following thresholding criteria to decide whether to insert a particle between two neighbors i, j in time step t:

- *Distance based*: if $||p_i^t - p_j^t|| > \epsilon_d$, with ϵ_d being a threshold which is a certain fraction bigger than the initial distance between two newly generated particles, insert p_k^t at $p_k^t = \frac{1}{2}(p_i^t + p_j^t)$.
- *Angle based*: if $\angle(p_i, p_j, p_k) > \epsilon_a$, insert particles p_r^{t-1}, p_s^{t-1} at $p_r^{t-1} = \frac{1}{2}(p_i^{t-1} + p_j^{t-1})$ and $p_s^{t-1} = \frac{1}{2}(p_j^{t-1} + p_k^{t-1})$.

Angle-based refinement is only performed on segments, whose length exceeds a minimal user-defined threshold value. While these criteria adjust particle sampling in a correct manner, they are not able to guarantee a certain accuracy with respect to the true streakline in flow fields with low resolutions on the time axis. In these cases, accuracy can be increased by inserting particles in a prior time-step $t - n$ if distance or smoothness criteria are violated in time-step t. We detail and extend these refinement techniques for the three-dimensional case as detailed later in this chapter.

Deletion of Particles As for particle insertion, we define distance and angle-based threshold for particle removal. We delete a particle p, if the total distance to its neighbors falls below a given threshold (e.g: $\frac{1}{2}\epsilon_d$) and the angle enclosed by its neighbors is greater than 120°, i.e. we do not lose details in the geometric representation. Choice of these thresholds comply with the rule of conservativity given in Section 6.3.

6.4.1.2. Boundary Integration

Another reason for us to propose grid-less methods for vector field approximation is the property of MLS to allow integration of stream- and streaklines on boundary geometry with no-slip condition, as velocities approximated this way generally do not evaluate to zero at the boundaries of the dataset. This behavior is desired when a true segmentation of the vector field is to be obtained and this form of approximation is compatible with the interpretation of simulation results. A major challenge during streakline integration is the process of boundary fitting, i.e. the event in time, when parts of a streakline first meet with geometry of the boundary object as illustrated in Figure 6.1.

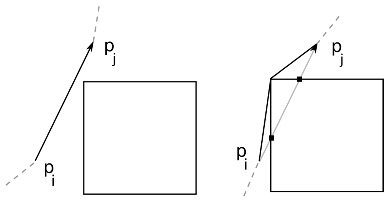


Figure 6.1.: A segment of a streakline is fitted to boundary geometry, after an intersection was detected.

Our fitting algorithm works as follows:
If a segment $p_i^t p_j^t$ of the streakline detects an intersection with boundary geometry, one of the following cases might occur. a) both particles penetrate the boundary of the dataset b) one particle penetrates the boundary object c) no particles penetrate the boundary object (as shown in Figure 6.1). In cases a) and b), the particles that penetrated the boundary surface are relocated to the intersection between their paths and the boundary surface and advanced to the appropriate updated position and the algorithm proceeds with case c) if there is no direct line of sight between these new positions. In case c), the intersection points of the line $p_i^t p_j^t$ with geometry are examined. If these intersection points lie on neighboring boundary elements, the common vertex is added to the streakline. If this is not the case, a new point is inserted half-way between p_i^{t-1} and p_j^{t-1} in the previous time step, and the algorithm is repeated for the new left and new right segment at the current time step. In our tests, this algorithm converged in less than 3 iterations.

A more straightforward approach might consider connecting the two points according to the shortest connection along the boundary object with respect to their corresponding intersection points, without inserting a new point in the previous time step. This however does not always produce correct and unambiguous results, especially in the case of big time steps and obstacles with a low resolution.

The movement along boundaries is performed as described in [OHBKH09b], with the additional requirement to keep track of the fraction of the current path length spent on the boundary object to correctly advance each particle to a new position on the boundary. Particles on boundary are released, if their velocity is virtually parallel to the boundary's normal, thus indicating a separating behavior.

6.4.1.3. Field Segmentation

The presented methods allow the generation of streaklines at user-specified positions throughout the dataset. Their capability to integrate along boundary makes them capable to indicate separating or, if integrated backwards, attaching behavior along their path, thus avoiding the necessity to track and match separation and attachment points throughout the whole dataset to obtain seeding locations for separating lines, when initial material boundaries are known. For specified initial material boundaries, streakline geometry represents the evolution of material interfaces over time and provides a time-varying segmentation of the flow field. This notion allows us to perform image-based domain segmentation for visualization purposes.

For this matter, we enclose the data set by a texture of arbitrary but sufficient resolution and use Bresenham's line algorithm [Bre65] to plot both the boundary of the dataset and the traces of streak- and streamlines. In a second step, all empty areas to the left and right of a streak- or streamline are filled and colored in a distinct color, using a simple recursive 4-neighborhood seeding algorithm. The dataset is then overlaid with the resulting texture, visualizing a segmentation into areas of similar flow (see Figure 6.11) or materials, termed *streak areas* in the following.

6.5. Adaptive Streak Surfaces

The following sections generalize and extend the notion of adaptive streak geometry in time-varying flow fields to three-dimensional fields.

6.5.1. Streak Surfaces

In the three-dimensional case, stream surfaces are replaced by streak- or timeline based surfaces [GKT$^+$08], which are still the topic of active research, as adaptivity measures during integration require an immense amount of additional processing, since a local approach to adaptivity as in [Hul92] is no longer feasible. In contrast to stream surfaces, streak surfaces are generally no longer tangential to the flow field and need to be updated or refined at their whole range during integration. Streak surfaces describe a truly three-dimensional complex in time and space, whereas stream surfaces are only two-dimensional in space. This increase of complexity prevents the use of classic approaches to adaptivity such as the concept introduced by Hultquist [Hul92].

Figure 6.2 illustrates a sequence of triangulations of three consecutive time steps of a streak surface. At the rake shown in blue, 15 particles are seeded at equidistant positions. The problem exhibited by non-adaptive approaches can be identified as the far too uniform particle distribution, preventing the accurate and smooth representation of folds. One has to note that in the discrete case a single streak surface consists of a number of consecutive static surfaces obtained in different time steps of the surface. Depending of the time resolution of the data set, coherency between consecutive surfaces might be low, producing a rough, jagged animation during rendering of the streak surface.

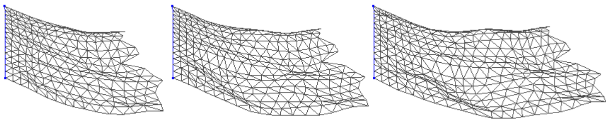


Figure 6.2.: Triangulations of a non-adaptive streak surface in three consecutive time steps.

6.5.2. Surface Generation

6.5.2.1. Algorithm Outline

Our algorithm to generate adaptive streak surfaces consists of five basic stages that are repeated for every time step of surface integration:

1. Generate new particles at the rake and insert them into the existing mesh
2. Concurrently advect all particles of the surface to their new positions
3. Restore the Delaunay property of the surface mesh by edge flipping
4. Determine curvature of the surface at every surface particle by MLS approximation
5. Adapt resolution by insertion and advection of new particles in current and previous time steps

These steps are described in detail in the following.

6.5.2.2. Particle Birth

Let $t \in [t_0, t_1]$ be the current time step. The basic representation of a streak surface usually consists of a set of particles that are advected through the flow field, being seeded at a predefined rake. Based on a user-defined resolution, we release a set of particles at equidistant positions along the rake. While the given resolution does directly influence the number of seed positions at the rake, the magnitude of the velocity field indirectly governs the number of particle rows that are generated and advected at the rake. If n particles with a seed distance of d are seeded at the rake, where the distance traveled in one time step at an arbitrary position on the rake is $l := ||f(.)|| \cdot \Delta t$, we release $m = \frac{l}{d}$ particles at every seeding position, leading to a total of $n \times m$ new particles.

Let T be a given triangulation in time step t of the particles released in $[t_0, t)$. We link the $n \times m$ new particles based on their neighborhood in parameter space (s, t) and connect this new triangulation to the first row of particles in T, as shown in Figure 6.3.

Consequently we obtain a fully connected particle-based streak surface, whose triangular mesh is used for surface approximation, refinement and visualization of the surface, as explained in the following sections. As a result, particles on a streak surface do not only carry spatial information, but provide data about the (s, t) parametrization of the surface, that can be used for the generation of texture coordinates, as well as normal and neighborhood information. It is important to point out that the basic definition, computation and visualization of a streak surface does not require the availability of connectivity information between particles. As described in the next sections, the triangulation created during particle birth is an auxiliary construct to reduce the computational effort for surface approximation and particle density calculations. While it only represents a linear approximation of the true streak surface, it can additionally be used for basic surface visualization.

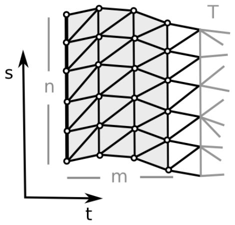


Figure 6.3.: The triangulation of the recently seeded particles is linked to the old mesh. Note that the old mesh is still located at its position in time step $i - 1$ until corresponding particles are advected.

6.5.2.3. Particle Advection

The particle advection scheme introduced in Section 6.2 reduces path deviations in data sets with large time steps and improves visual coherence. A mapping of the order of integration k onto a streak surface is shown in Figure 6.4. As individual particles from surfaces of consecutive time steps are matched in our data structure, storing not only the final position of an advected particle, but reusing the k intermediate ones obtained from field integration during streak surface visualization facilitates the rendering of smooth surface animations even in data sets with low time resolution. In these computations, the magnitude of the smoothing length r of the weighting function used in MLS during vector

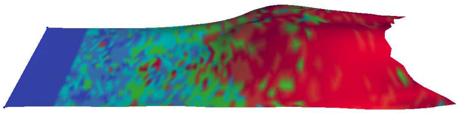


Figure 6.4.: A simple streak surface is color-mapped with the integration order $k \in [1, 16]$. Order of integration is mapped to the hue spectrum from $0°$ to $240°$, with red being the maximum. As can be seen, even adjacent particles might need a highly different number of vector field evaluations.

field evaluation is inversely related to the point density of the data set. This point density is usually either an output value of the CFD simulation itself, or has to be determined on the fly by k-nearest neighbor computations or similar methods. We use MLS for vector field approximation because of its independence of a computational grid and because it is used as interpolant by the simulation that generated our test data sets.

To speed up the advection process, particle locality is used for efficient data set caching and parallel particle advection. We impose a rectangular grid on the surface clustering particles into independent sets and delegate the according computations from (6.1) to different CPU cores.

If a particle of the surface leaves the boundaries of the data set during advection, the particle itself and adjacent mesh elements are deleted, efficiently trimming the streak surface.

6.5.2.4. Delaunay Meshing

As mentioned before, the increased dimensionality of streak surfaces has a direct influence on the adaptivity methods that can be used. Hultquist's approach of inserting streamlines at a single curve-like consistent front cannot be generalized to the insertion of streaklines at a one-dimensional front of the streak surface. The two-dimensional character of the generalized front definition for streaklines requires the insertion of refinement structures on arbitrary positions of the surface. This fact virtually rules out the exclusive refinement along time- and streaklines of the surface, as most interior refinement points would miss these grid lines. Thus, we use a Delaunay type progressive mesh to define criteria for particle insertion.

Various work on the construction of Delaunay type surfaces from point clouds has been published over the years. For example, Gopi et al. [GKS00] use a projective local Delaunay mesh for surface reconstruction. The underlying particle concept of streak surfaces suggests the use of point cloud reconstruction methods to obtain a surface mesh. However, the evolving property of the streak surface mesh facilitates topologically correct (re-) meshing in a certain time step based on connectivity information given by prior time steps as well as triangle, edge,

and node matching over multiple time steps.
The mesh structure of the previous time step generally loses its Delaunay properties as particles are advected, if corners of adjacent triangles describe different paths through the flow field, as illustrated in Figure 6.5. Since we want to estimate particle distributions using circumcircle properties for mesh refinement in a later step of the algorithm, it is important to have a well conditioned triangulation avoiding skinny triangles. Therefore, we choose to impose Delaunay's mesh properties on the triangulation of our surface, as the minimal circumradius property is a direct indicator of particle density. The availability of a mesh on the current particle set greatly simplifies construction of a curved Delaunay mesh, since this fact makes it possible to use local edge flipping instead of global mesh construction algorithms such as line-sweep. Moreover, reusing the previous particle connectivity allows matching of non-flipped edges as well as triangles of different time steps.

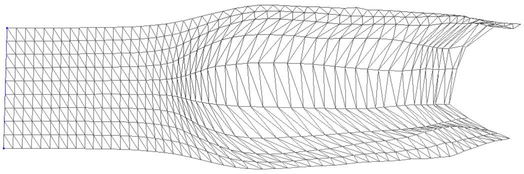


Figure 6.5.: A uniform grid of a non-adaptive streak surface gets deformed, producing ill-conditioned triangles of bad aspect ratios.

An edge of a mesh is *flippable*, if it is shared by two triangles and its flipped counterpart is not already part of the mesh , see also [DZM07]. Let $e = (b, c)$ be a flippable edge shared by the two triangles $\Delta_1 = (a, b, c)$ and $\Delta_2 = (d, c, b)$. We flip e, if the sum of the angles α at a and β at d exceeds π, resulting in $\Delta'_1 = (a, d, b)$ and $\Delta'_2 = (c, d, a)$, satisfying local Delaunay properties. This flipping procedure, having been shown to converge for curved surfaces by Dyer et al. [DZM07], is repeated until the mesh contains no more flippable edges. If non-flippable edges of tetrahedral structures of the curved triangle mesh remain after all flippable edges have been swapped, see Figure 6.6, we collapse the corresponding tetrahedra by removing the particle at its tip. Such tetrahedra generally represent noise in the form that they indicate a particle that evades the path of the streak surface and has earlier been inserted at a wrong position. After this step, our surface mesh is usually Delaunay conform, with exceptions to rare special cases of non-flippable edges. We avoid insertion of additional particles on the surface to obtain a mesh that fully satisfies Delaunay properties as proposed in [DZM07] to reduce the resulting computational overhead that is needed to advect the new particles through the vector field. In these special cases we allow our mesh to be locally non Delaunay, as the according triangles are commonly not badly shaped due to the fact of mesh deformation of large parts of the surface in every step of integration.

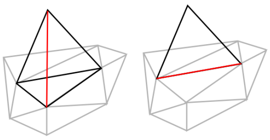


Figure 6.6.: Edges of a tetrahedral structure on the curved surface mesh cannot be flipped without changing the underlying topology and creating holes.

6.5.2.5. Curvature Approximation

We determine local geometric complexity of the streak surface by measuring its local curvature. Curvature at a point p of a bivariate surface S is described by the maximal and minimal curvature of the curves on S that result from intersecting S with planes through p containing the surface-normal vector at p. These curvature values are called *principal curvatures* and is used in the following as an indicator of whether to refine the mesh of the surface.

For adaptive particle insertion we need to know both a parametric form of the surface, as well as its local curvature at every particle of the surface. Instead of handling these tasks by two different approximation techniques, we use one weighted Least Squares approximation to both obtain a valid surface representation as well as to calculate the according local curvature. We compute local surface approximation at a particle p based on particle offsets from a local tangential plane with vertex normal n_p as shown in Figure 6.7, resulting in a new set of projected data points $(p_i' = (x_i, y_i), d_i)$ that are approximated by a scalar-valued bivariate quadratic MLS. If the surface is to be approximated at a particle p, we assemble surrounding particles p_i by a neighborhood search along the edges of the mesh up to a neighborhood distance of two, as the bivariate quadratic LSE that has to be solved for a MLS approximation requires the data of at least six non-collinear points. The smoothing length r of the weighting function used during MLS approximation is chosen in a way, such that $w(p, p_j') = \epsilon$ with p_j' being the neighboring particle that is farthest away from p. This MLS surface representation defines every point on a surface by its orthogonal distance to the

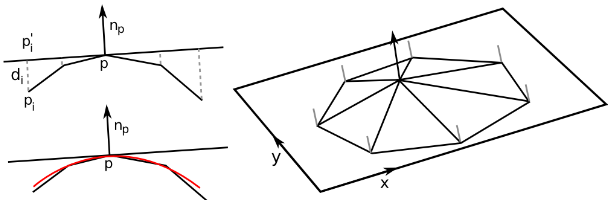


Figure 6.7.: Two-dimensional illustration of a MLS curve approximation, MLS curve is shown in red (left) and three-dimensional analogon with a particle neighborhood-level of one (right).

tangential plane.
The eigenvalues of the Hessian of this polynomial approximation represent the principal curvatures c_1 and c_2. The Hessian of a bivariate scalar function takes the form of a 2×2 matrix, which can easily be calculated using a parametric form of the approximating MLS polynomial [KK06]. We use the *maximum absolute curvature*

$$c = \max(||c_1||, ||c_2||)$$

as a measure of local feature size, which has proven to lead to good results in other applications, see [dAJ04]. Figure 6.8 shows a simple streak surface colored according to maximum absolute curvature. While Moving Least Squares for surface approximation is computationally more expensive than to simply use the piecewise linear representation of the surface described by the mesh itself, it both reduces errors in the particle insertion step, as explained in the next section and gives a more accurate notion of the surface curvature, by filtering small noise due to its approximating behavior.

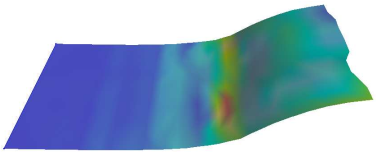


Figure 6.8.: Maximum absolute curvature as hue color-map on a streak surface. Red color shows regions of high curvature, blue indicates planar sections.

6.5.2.6. Particle Insertion

The general notion of adaptivity is to sample a surface according to its geometric complexity, meaning that regions of high curvature need more samples to be represented accurately than regions that are almost planar. In the context of parametric or implicit surface modeling, curvature-dependent particle density control is often handled by minimization of an energy function [MGW05]. In our case the availability of a correct coarse mesh as well as the absence of a gradient-definition requires different adaptivity measures.
The crucial step of streak surface adaptivity is the correct insertion of new particles at positions throughout the surface. The most accurate way of adaptively inserting particles is the insertion of a particle in the appropriate first time-step t_0, whenever an ill-conditioned particle-density is detected at an arbitrary later time step t_1. This method does however require the computationally inefficient advection of all newly inserted particles from t_0 til t_1. We therefore prefer to insert new particles directly into the evaluated time step or few of its predecessors. Given the computed curvature at a particle p, we are able to decide whether to insert new particles in the immediate neighborhood of p. Figure 6.5 demonstrates,

why refinement purely along time- and pathlines is not desired in the context of streak surfaces, since right angles between such lines are not maintained over time and such a skewed global coordinate frame is not suitable for balanced control of particle densities. We therefore use the local feature size in form of the curvature at a position p to directly describe the desired maximal allowed distance between two neighboring particles:

$$r_c = \frac{b}{\max(||c_1||, ||c_2||)}$$

where b controls the impact of curvature on the degree of mesh refinement and is commonly chosen be a value between zero and one. Circumradii of all triangles adjacent to p are compared to r_c, if a circumradius exceeds this threshold, and is larger than a pre-defined minimum triangle size, a new particle is inserted on the according triangle. We commonly limit the minimum triangle size to a size a few magnitudes smaller than the data set resolution to avoid numerical instabilities and oversampling. As our underlying mesh is mostly Delaunay conform and therefore satisfies the smallest circumcircle property, the circumradius of triangles is a valid measurement of particle density at a specific region.

Particle insertion itself poses the question of how to find the optimal position of insertion. On planar Delaunay meshes, one would choose the circumcenter of a triangle as location and handle point insertion in a simple way: Remove all triangles, whose current circumcircle includes the newly placed point and connect vertices of the resulting polygon with the new point. The resulting mesh again satisfies the Delaunay property. However, on curved surfaces this method yields several problems. Location of the circumcenter of obtuse triangles is not trivial, as the circumcenter is located outside of the triangle, thus requiring a search on neighboring triangles and even performing intersection operations, if the surface bends. Moreover, the point-in-circumcircle property does not work as expected from the planar case for both the common projected-circumcircle and the circumsphere definition on curved surfaces, if the point is not inserted on the piecewise linear representation of the surface. These considerations motivate our approach of particle insertion.

Once a triangle with a too large circumradius is detected, the according new particle is either inserted at the centroid of this triangle or on one of its edges. More precisely, we subdivide the longest edge of the triangle if the circumcenter is located outside of the triangle or the triangle itself is a boundary element of the surface. The problem caused by an invalid insertion of particles at the centroid of a boundary triangle is depicted in Figure 6.9, where the quality of the aspect-ratio of the boundary triangle is degraded significantly. These two types of particle insertion lead to well conditioned triangles with smaller circumcircle size as soon as the Delaunay property of the mesh is restored in the mesh at the next time step.

This insertion scheme yields adaptive streak surfaces but tends to introduce

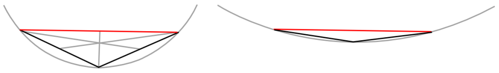


Figure 6.9.: Circumradius of the triangle adjacent to the boundary edge (red) increases after particle insertion, as the boundary edge cannot be flipped to produce better conditioned triangles. Such cases are avoided by splitting of boundary edges.

errors in the particle insertion positions that lead to further surface deviations during advection, even if newly inserted particles are offset according to the local MLS approximation of the surface as computed in the previous step of our algorithm. To avoid these artifacts, we introduce the concept of looking back at a history of $lb > 0$ previous instances of concerned triangles or edges. If a triangle (edge) tri_1 is bound to be refined in time step t_1, we find matching triangles (edges) in time steps $t_i \in [t_1 - lb + 1, t_1 - 1]$ and insert particles into according triangles in t_i, until either step $t_1 - lb + 1$ is reached, the curvature of the surface at the corresponding position in t_i falls below a given minimum, or no matched preceding triangle exists and cannot be created by edge flips. These m newly inserted particles describe an ordered sequential set of points $[p_0, p_{m-1}]$ located on the linear representation of the individual surfaces $[S_{t_1-m}, S_{t_1}]$. As these particles need to lie on the same pathline, positions of particles p_i, with $0 < i \leq m-1$ are offset according to the position of p_{i-1} after one full time step of particle advection.

Our tests have shown that this leads to a great reduction of noise artifacts in later time steps, that are caused by deviating paths of particles that were inserted at poorly approximated positions.

Curvature as well as circumradius properties can be used for the removal of particles as well. In the case that a particle p has a sufficiently small curvature, i.e. the surface is almost planar, and all adjacent triangles have small circumradii, it is valid to remove the concerned particle without losing any details in the surface representation. In realistic applications such as the test data sets shown in the next section such situations do hardly occur.

6.6. Results

Figure 6.10 shows the benefits of streakline adaptivity measures in a planar dataset with approximately 3.000 points. Significant improvements in streakline accuracy can be observed. Area deformations near the same flow obstacle are shown in Figures 6.11 and 6.12. Initial starting positions for streaklines were set manually in a symmetric fashion. In practice, these points are defined by the input material distribution and should be an application specific output of the simulation. The inner area in the periodic flow of Figure 6.12 is caused by a single boundary-integrated streakline, which was started in the middle of the entry region of the simulation. As indicated by a comparison of the Figures 6.11 and 6.12, such streaklines allow a more detailed view of the mixing process. Figure 6.13 demonstrates the benefit of flow segmentation to multi-field visualization. Visual complexity of scalar field visualization is reduced by selective application of transfer-functions. Scalar values are again mapped to hue in HSV space. This benefit becomes more evident in three-dimensional data sets as shown in Section 6.5.1. The main drawback of boundary streaklines is the computational cost of their integration, since during every integration step intersection tests with boundary have to be computed and particles are traced along boundary elements. The factor of increased computation time depends strongly on the complexity of the boundary object. In the shown datasets, a factor of less than 2 was observed.

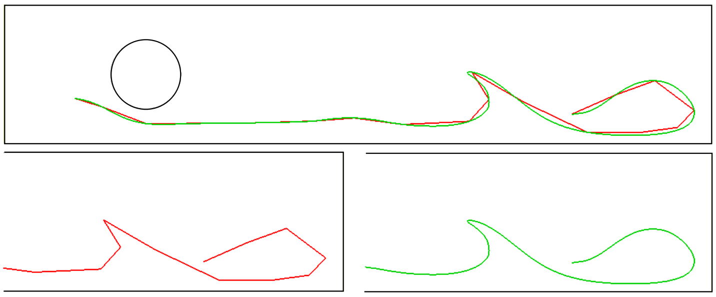


Figure 6.10.: Comparison between non-adaptive (red) and adaptive streakline (green) with identical point seeding frequency. Angle- and distance-based refinement criteria improve streakline accuracy in regions with high turbulence.

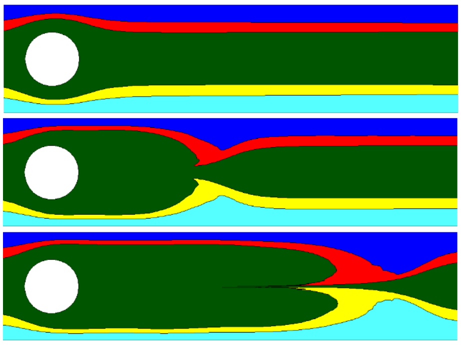


Figure 6.11.: Visualization of streak areas in a sequence of three time steps. Bending, stretching, thinning and thickening deformations in all areas are visible. Area analysis near the right border of the dataset provides information about output material distributions.

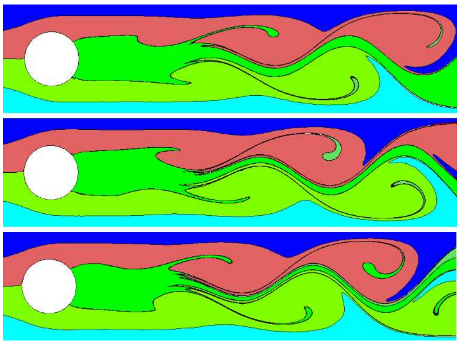


Figure 6.12.: Visualization of periodic streak areas in a Karman vortex street with boundary streakline. Periodicity, frequency and form of the deformations give an impression of the final material composition.

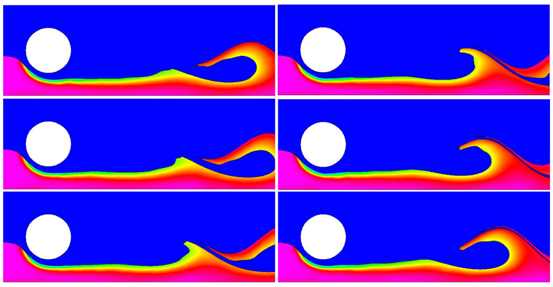


Figure 6.13.: Segmentation allows multi-field visualization by the direct visualization of scalar quantities such as vorticity.

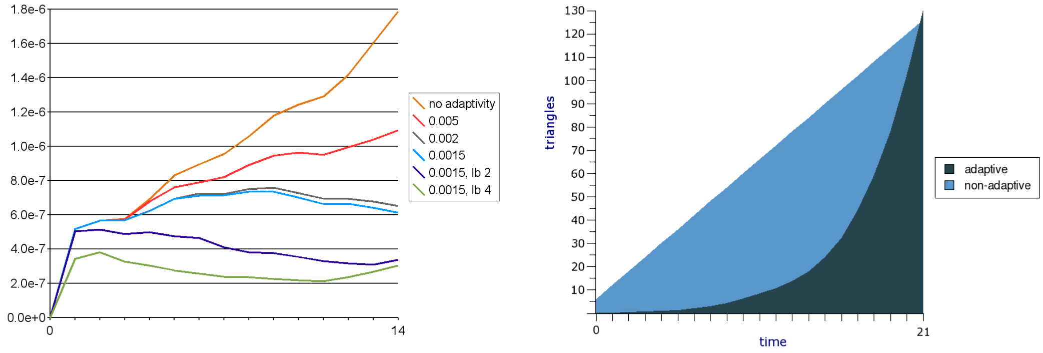
In the following we present numerical examples of adaptive streak surface integration in different data sets generated by a point based, grid-less CFD simulation [TK02]. The first data set consists of about 25.000 particles simulating flow around a cylindrical obstacle with ellipsoidal profile. Dimensions of the obstacle as well as velocity of the fluid were chosen specifically to yield a high Reynolds number, leading to a three-dimensional Von Kármán vortex street and a low time resolution. The resulting disturbed flow structures have optimal properties to observe the quality of our adaptive streak surface integration approach. Figure 6.15 depicts six non-consecutive time steps of adaptive streak surface integration in this first data set. Near the vortex structures, the surface mesh has a high curvature and needs to be sampled accordingly. Highly twisted and folded surface structures in the Von Kármán vortex street demonstrate the robust refinement of our approach. The surface in the last time step shown consists of around 110.000 particles.

The second data set contains a spherical obstacle, approx. 26.000 flow particles and fluid flow with a Reynolds number close to the one from data set one. Both data sets have a low resolution in the time dimension to demonstrate the robustness of artifact avoidance by utilization of our look back strategy. In contrast to the first data set, absence of a distinguished direction on the obstacle leads to aperiodic, intensively folded flow structures, see Figure 6.16. To give an impression of how accurately surfaces generated by the adaptive integration scheme and the Delaunay mesh refinement method introduced in this work represent the "real" surface, we show comparisons between high-resolution non-adaptive meshes and adaptive surfaces in the two Figures 6.21 and 6.22. The former illustrates how the folds of a high-resolution mesh are correctly refined and represented by an adaptive surface with only half as many triangles as an equivalent non-adaptive surface. Efficient distribution of particle densities can be observed in Figure 6.22, where the adaptive version of a streak surface clearly has a particle distribution of better quality than a non-adaptive surface with an equal number of points. Figure 6.17 shows an example of flow segmentation performed by surface voxelization as described in Appendix B. This shows, how streak surface based flow segmentation contributes to multi-field visualization by allowing selective application of transfer functions. We emphasize that generation of adaptive streak-surfaces followed by subsequent voxelization usually generates more accurate volume representations than comparable texture based advection techniques. The techniques for strain computation and visualization presented in Chapter 5 can be used for multi-field visualization and analysis of relative and absolute strain directions along the streak surface, as shown in Figure 6.18. Statistical error measurements are shown in Figure 6.14a. The measured two-sided surface mesh error is computed with respect to (6.2).

$$E(S_1, S_2) = \frac{1}{|P_1| + |P_2|} \left(\sum_{p \in P_1} d(p, S_2) + \sum_{q \in P_2} d(q, S_1) \right) \tag{6.2}$$

where S_1 and S_2 are the meshes of two streak surfaces at the same time step with according point sets P_1 and P_2. It is important to note that, as d computes the minimal distance of a point to a surface mesh, error values for the same point sets usually differ if the mesh is changed. While small features in the error curves of Figure 6.14a may therefore be caused by flipped edges, the overall benefit gained from reducing the minimal surface resolution and increasing the number of look back steps is clearly visible. The plotted exemplary measurements were taken at different time steps of a data set that was scaled to fit into a unit cube and represent absolute per particle error when comparing the according surface to a high resolution ground truth streak surface. The shown measurements represent 15 consecutive time-steps of a surface evolving under extreme stretching and turbulence conditions - a representative scenario that requires reliable surface reconstruction techniques to produce accurate streak surfaces.

In addition to the direct impact of our adaptivity approach on particle counts and distributions, the reduced number of particles does influence the computation times of surfaces as well. We show a representative graph of particle number development in a surface generated in the ellipsoidal data set in Figure 6.14b.



(a) (b)

Figure 6.14.: (a) Error statistics for different levels of adaptivity and look back. (b) Chart of the number of triangles (in multiples of 1.000) in the first 22 time steps of the streak surface shown in Figure 6.15. Dark blue refers to the adaptive streak surface, light blue to the non-adaptive streak surface, that has an approximately identical number of triangles at step 22.

Method	Advection	Delaunay	Adaptivity
non-adaptive	1.768.470	2.865	0
adaptive, 0.002, 2 lb	790.890	1.155	9.081

Table 6.1.: Exemplary measurements of total streak surface generation times in milliseconds for 25 time steps of data set one with 20× parallel particle advection. Final particle number is approx 40.000 for both approaches.

The triangle count for the adaptive surface shows exponential growth as long as the surface is completely inside the domain of the data set and is not trimmed. Exponential growth is stopped as soon as parts of the surface leave the data set. Integrals of the depicted curves are directly proportional to the time spent for total particle advection. From the given graph, it is clearly visible that adaptive streak surfaces require the advection of a larger number of particles to reach the same final number. Due to better distribution of particle densities, an adaptive surface with the same number of particles generally represents a better approximation of the real surface. The previously mentioned assumption about particle advection time is verified by measurements during surface integration. The total time spent for adaptivity measures is a combination of computation times for surface and curvature approximation as well as particle insertion. In our tests only a percentage of less than 2% of the total surface generation time was consumed by the adaptivity method, even if particle advection was performed 20 times in parallel. As adaptive integration schemes commonly requires multiple evaluations of the vector field for every particle, the amount of time dedicated to particle advection is much higher than the one used up by mesh refinement, thus noticeably speeding up the integration process. Further improvements on this performance can be made by parallelization of the mesh refinement algorithm. While in our tests generation of adaptive surfaces was in average 4 times faster than creation of equivalent higher resolved surfaces, when using 10× parallel particle advection, it is difficult to obtain meaningful absolute time comparisons (cf. Table 6.1), as speed-up is highly dependent on the method of vector field approximation and flow complexity. In general, the relative speed-up gained by adaptive surface generation is proportional to the complexity of flow and field evaluation methods.

For insightful visualization of generated surfaces, we utilize several known visualization techniques. Four of these techniques are displayed in Figure 6.19. While the first picture showing a standard particle based visualization of the surface is not capable of conveying the impression of a homogeneous surface, it gives an insight into adaptive particle distributions. Particle density and connectivity information are shown by a direct wire-frame representation of the Delaunay construct as used in the second frame. Shading and texturing techniques are applied to the solid Delaunay triangulation based surface visualizations. We use a simple axis-based triangle pre-sorting approach for transparent surface rendering as shown in the last two frames. Texturing allows the rendering of streak- or timeline like parts of the surface, former is shown in frame four.

These renderings of streak surfaces have in common that the Delaunay construct used for density estimation is used directly for surface visualization, leading to a simple and fast visualization allowing the analysis of adaptivity properties. For high quality surface rendering, one would typically use a smooth interpolation of the surface point set and a much finer resolved triangle grid to get rid of fine discontinuities of the simple linear representation.

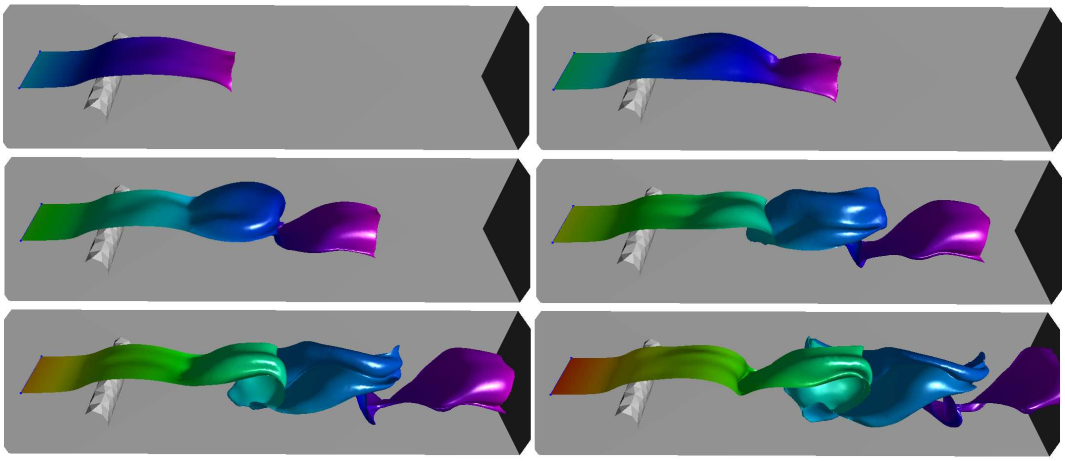


Figure 6.15.: A sequence of six time steps in a three-dimensional Von Kármán vortex street produced by test data set number one. The extracted surface is colored with respect to time, with purple showing the particles generated at time t_0 and red being used for new particles. Details of twisting folding, and stretching are visible.

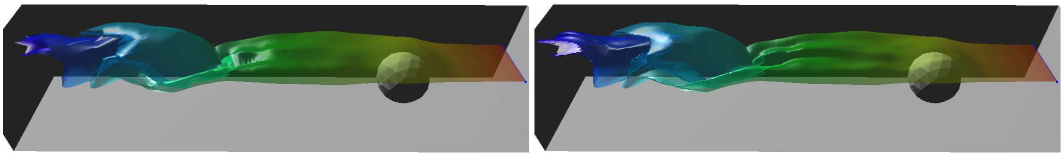


Figure 6.16.: Transparent rendering of a non-adaptive and an adaptive streak surface with the same seeding resolution extracted from test data set number two. The spherical obstacle produces aperiodic disturbances.

To illustrate the use of the methods introduced in this chapter in a real application, we present an example of an adaptive evolving mesh being equivalent to a streak surface without a seeding rake in Figure 6.20. The application is concerned with stirring of a fluid at high temperature and consists of a cylindrical barrel and two rotating wheels with four attached mixing poles. The extracted mesh segments the fluid in a stirring simulation into two separate volumes, giving an impression of how our approach to adaptive streak surfaces can be used to separate regions of different flow as done in vector field topology.

6.7. Summary and Discussion

This chapter has introduced methods for adaptive modeling of macroscopic flow deformations. The proposed streak areas and streak surfaces facilitate accurate segmentation of time-varying flow fields for two- and three-dimensional fields.

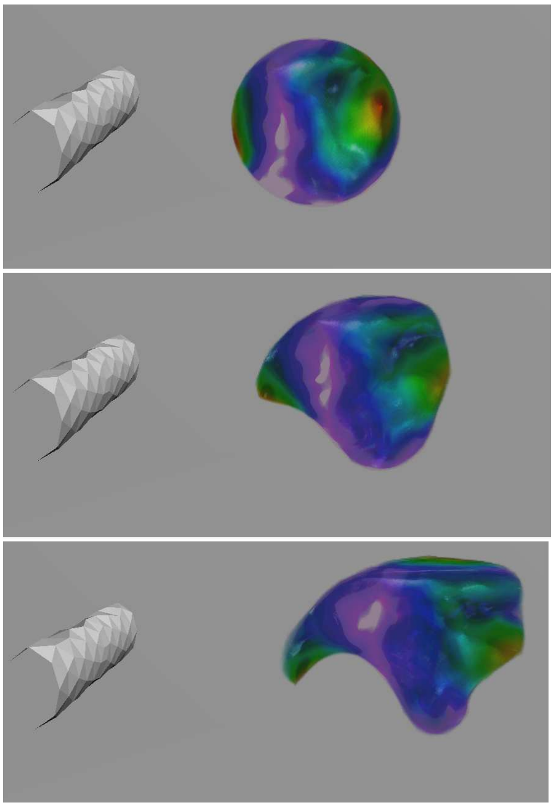


Figure 6.17.: Volume voxelization of a closed deformable mesh is used for volume rendering of flow vorticity magnitude (mapped to hue). Such flow segmentation allows multi-field visualization. Flow segmentation by adaptive meshes allows selective application of transfer functions to scalar-valued flow quantities. Observed extremal values of vorticity at the boundary of the segmentation can be seen to be related to mesh deformation in later timesteps.

While adaptivity measures reduce the amount of expensive field evaluations and particle numbers, especially the three-dimensional case requires robust geometry processing. Flow field segmentation reduces visual complexity of direct scalar field visualization.

Streak-area generation inherently avoids intersection of streaklines and bypasses the problem of entry/exit point tracking by the integration of streaklines on boundary geometry. An advantage of segmentation techniques is the possibility to use segmentation information in a multi-field visualization context. The adaptivity and integration methods developed in this section can be generalized to advect arbitrary material boundaries through the flow field.

Furthermore, we have introduced an approach to fully adaptive streak surface integration. Our method is based on MLS approximation for the evaluation of the vector field and surface as well as curvature approximation. For particle insertion

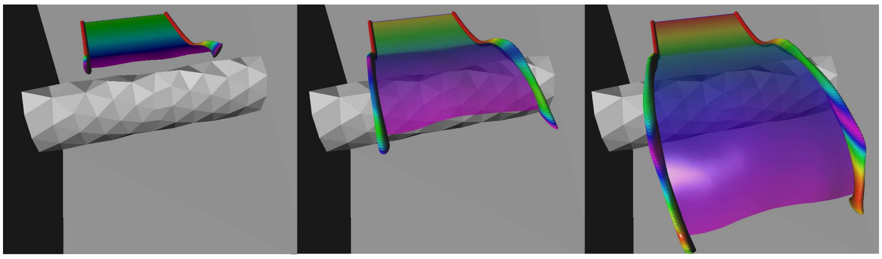


Figure 6.18.: Strain integration along streak-surface border reveals strain direction. Resulting multi-field visualization indicates high strain orthogonal to the streak surface after the flow obstacle. High strain values highlight regions that require highly accurate particle advection to avoid surface artifacts. The shown streak surface is colored with respect to seeding time.

we utilized the circumradius properties of a Delaunay type evolving mesh, which is restructured after every time step to restore Delaunay properties. The distinction between two types of particle location during insertion avoid the generation of triangles with bad aspect ratios. A look back strategy during particle insertion further reduces approximation errors. The results demonstrate the accurate, robust, and fast integration of adaptive streak surfaces generated by our method. Moreover we have presented results in a real world application of a mixing process, illustrating the suitability for practical flow analysis and topology-based visualization. The methods introduced in this work are portable to other fields of visualization where moving or evolving meshes are concerned, such as the concept of unsteady flow volumes presented by Becker et al. [BML95]. Common approaches to the extraction of vector field topology of fluid flows [WTS09] based on the use of stream surfaces as three-dimensional separatrices in the stationary case can be generalized to vector field topology methods in non-stationary fields by the application of streak surfaces for segmentation of time-varying volumes.

Future work on the topic of adaptive streakline and surface construction may include the utilization of improved spatial clustering for further parallelization of the generation process as well as the integration of vector field singularities into the streak surface generation process. Furthermore, the ability of streakline and surfaces to sample flow fields accurately is an interesting property that can be used for flow analysis and segmentation [OHBH10] and allows definition of dense flow statistics.

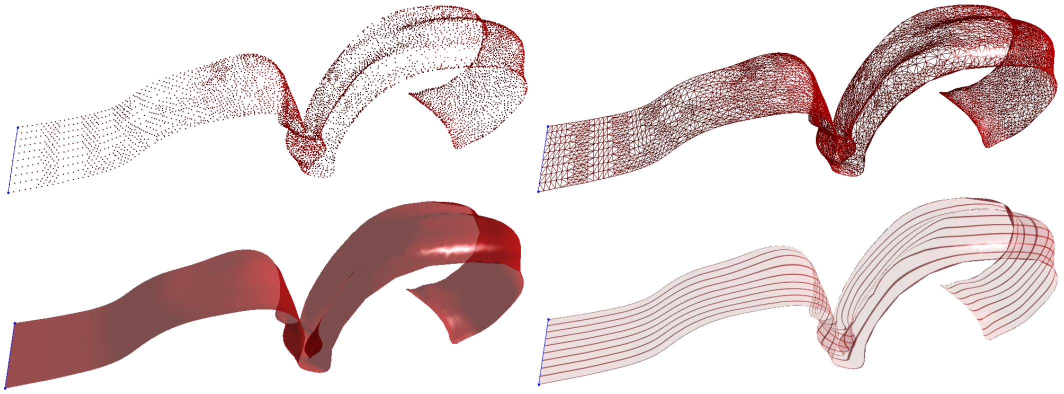


Figure 6.19.: One time step of streak surface integration shown in four different visualization techniques for surface rendering. Techniques are: Surface particles rendered as shaded points, wire-frame triangulation, solid transparent triangulation, and texture mapping showing streakline structures.

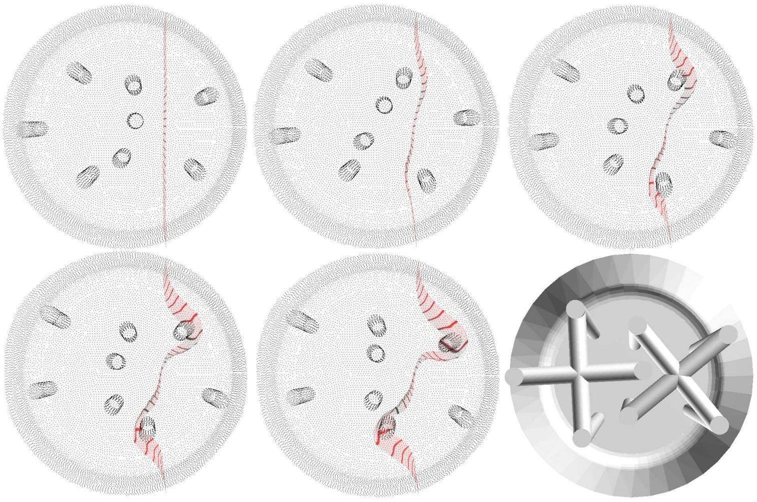


Figure 6.20.: Refinement mesh in a stirring simulation.

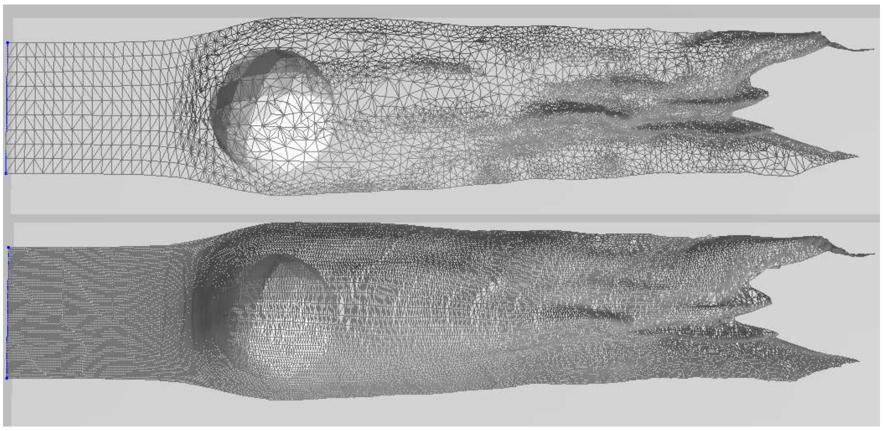


Figure 6.21.: Triangulations of an adaptive (top) and a high-resolution non-adaptive streak surface passing a spherical obstacle. Triangle counts in the time step shown evaluate to approximately 20.000 and 45.000. The adaptive surface yields a highly diverse particle density, while accurately representing fold-like features.

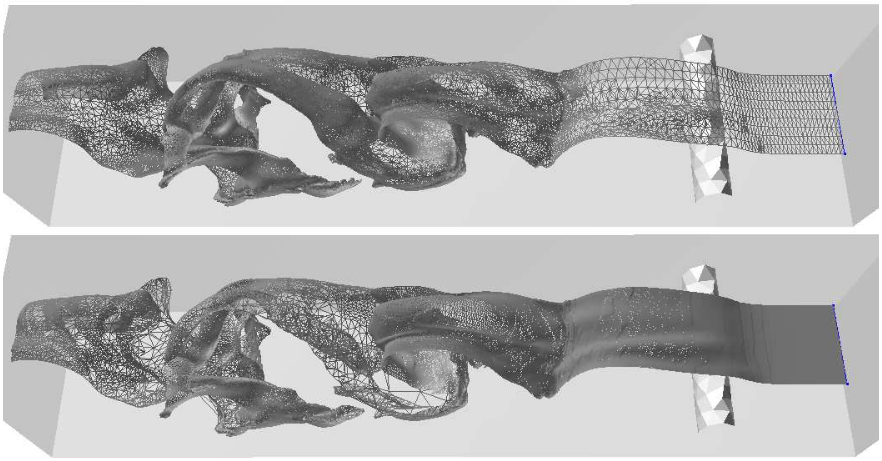


Figure 6.22.: Triangulations of an adaptive (top) and a non-adaptive streak surface with approximately 110.000 triangles each. The non-adaptive version shows less optimal particle distributions and regions with low mesh resolution.

7. Tensor Fields - Scattered Moment Tensor Data

Instead of being a derived quantity from flow simulations, the displacement data processed in this chapter is based on real-world measurements obtained during earthquakes. Data is available in the form of moment tensors which model surface displacement. In contrast to previous chapters, where visualizations of multiple fields are combined in one visual representation, the methods developed in this chapter use separate views of the same data set to convey multi-field data. In the following sections we are concerned with the visualization of scattered tensor data that represents moments in geophysical data sets.

Seismic earthquakes in geoscience are a comparatively less known source of tensor data when compared to applications from medical data visualization, such as DT-MRI. Due to the measurement technique, that places seismic sensors at locations close to points of interest such as known or suspected faults, the generated moment tensors are represented as scattered set of symmetric second order tensors describing earthquake point sources leading to displacement of the earth's surface.

This scattered tensor field carries valuable information for earthquake and fault analysis such as magnitude, type, wave polarity and fault orientation. We present novel tensor glyphs to visualize these properties and extract meaningful tensor features in individual moment tensors. Interpolation in moment tensor fields is a challenging task, as different source types, such as artificial explosions, natural shear dislocation, and volcanic eruptions have to be treated separately. To overcome this problem, we introduce novel moment tensor clustering and averaging methods as well as accompanying visualization techniques. Tensors of resulting clusters can be combined and averaged into one representative glyph, which additionally contributes to the quality of glyph-based field visualization by reducing visual occlusion and clutter. In our work, the spatial nature of the information conveyed by placing tensor glyphs at positions of measurements is complemented by a visualization of the overall orientation data present in the moment tensors in the form of a hemisphere stereographic projection. This allows us to depict clustering and orientation not only in three-dimensional but also in the projective space of orientations. Furthermore, we provide interaction methods to intuitively match data between both spaces. While the presented methods may be used for the illustration of stress tensors in general, they have been applied to different real world moment tensor data sets to produce the results presented in this chapter.

The main contributions of the following chapter to the visualization community are novel moment tensor clustering and averaging techniques, polarity glyphs and slip geometry for visualization of indefinite tensors, and linked spherical projective and spatial visualization techniques for scattered moment tensor analysis.

Section 7.1 provides an overview of existing work in the field of tensor visualization and moment tensors in special. In Section 7.2, we give an introduction to basic properties, types and sources of moment tensors. Section 7.3 proposes new glyph techniques to highlight important characteristics of moment tensors before introducing tensor based clustering and averaging in Section 7.4. We built upon these findings, to describe properties and interaction techniques of the two spaces used for visualization of scattered moment data in Section 7.5. Results are given in Section 7.6, whereas Section 7.7 concludes this chapter.

7.1. Related Work

Most tensor field visualization techniques [WH06] focus on either local properties of individual tensors or global behavior of the tensor field. Visualization of local attributes of the tensors is commonly achieved by utilizing shapes, icons or glyphs to depict relevant tensor properties, as done by superquadric glyphs [Kin04], Reynolds glyphs [MSM96] or others, whereas methods that need a global topology-based analysis of the field make use of streamline extraction along tensor eigenvectors [DH92]. These techniques usually assume a densely sampled field or uniform treatment of tensors. Most recently the use of superquadric glyphs for the visualization of indefinite tensors was proposed in [SK10]. However, the shape of these tensors alone is not able to convey positive and negative definite regions of the tensor. To depict moment tensors, Ohtsu et al.[OS04] use VRML and simple plane glyphs. These glyphs are limited to the display fault plane and movement directions. Another publication in the area of moment tensor visualization [NJP05] uses dense display of shapes on a regular grid to depict eigenvector directions and magnitude. However, they do not use explicit clustering or occlusion reduction techniques. In general, when tensor fields are sampled in a significantly dense manner or allow straight-forward interpolation of tensors, a global view of the data can be achieved by dense or intelligent glyph placing [HSH07]. In contrast to these methods, we aim at giving a global view of tensor properties in a scattered data set, while avoiding the drawback of classic tensor interpolation. In the context of clustering, Rohlfing et al. [RSP07] make use of diffusion-tensor k-means clustering based on probability distributions to allow averaging and interpolation between DT-MRI scans. Regarding moment tensors, similarity measures based on the inner matrix product [Wil93] have been used for simple cluster identification; derived clustering techniques are used in a semi-automatic manner in this work. Currently, the tensor visualization community has mainly focused on processing and analysis of diffusion tensors. We intro-

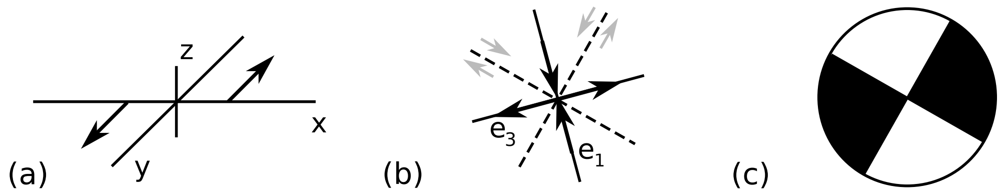


Figure 7.1.: (a) Vector couple corresponding to m_{yx}. (b) Force dipoles in rotated coordinate system corresponding to pure shear. The stippled lines represent the two possible fault plan orientations. (c) Standard beachball representation of the situation in (b).

duce methods that are not specific to diffusion tensors, but aim at improving moment tensor data analysis. However, the concepts introduced in this work are applicable to a variety of symmetric second-order tensor fields.

7.2. The Moment Tensor

Seismic sensors allow the recording and processing of waves emitted during earthquakes, indicating surface displacement along faults. A mathematically well-founded way of capturing these displacement discontinuities of the earth's surface is their representation as force couples in moment tensors [AR80], see Chapter 3. The seismic moment tensor M is a symmetric second order 3×3 tensor. Tensor components m_{ij} denote the magnitude of the moment caused by the force couple in i-direction acting on the the j-direction. Figure 7.1 (a) illustrates one of nine possible force couples. The vanishing angular moment in the seismic source implies symmetry, i.e. $m_{ij} = m_{ji}$. In the following sections we present fundamental properties of such moment tensors.

7.2.1. Fault Representation

An important information represented by moment tensors is the orientation of displacement discontinuities or fault planes in the earth's mantle. Due to symmetry, a moment tensor M can be diagonalized by rotation into its eigenvector system, with entries on the diagonal corresponding to its eigenvalues. For pure slip along a plane, M has eigenvalues $\lambda_1 < \lambda_2 < \lambda_3$ with $\lambda_1 = -\lambda_3$ and $\lambda_2 = 0$, indicating vanishing displacement in a direction orthogonal to a faulting plane, see (7.1). As only one force couple contributes to displacement, such moment tensors are therefore known to be *pure double-couple* (DC) tensors.

$$M^{DC} = \begin{pmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{pmatrix} = \begin{pmatrix} -\lambda_3 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \lambda_3 \end{pmatrix} \tag{7.1}$$

In the rotated principal axis coordinate system, where $-\lambda_3 = m_{xx}^{DC}$ and $\lambda_3 = m_{zz}^{DC}$, the corresponding eigenvector directions e_1 and e_3 represent two perpendicular force dipole directions of opposing orientation, as shown in Figure 7.1 (b). Due to displacement magnitude and direction, possible fault plane directions are rotated by 45° with respect to the principal axis directions. This results in two possible plane orientations, where one corresponds to the true fault plane and the other is an auxiliary plane. Only analysis of further seismic activities can help select the true fault plane direction from these two orthogonal candidate planes.

7.2.2. Moment Tensor Decomposition

Due to measurement errors, complex faulting behavior, and other sources, eigenvalues of real-world moment tensors do not always have the properties described in 7.2.1 and do generally not represent pure double-couple tensors. A tensor decomposition proposed in[KR70] allows measurement of the contribution of a pure double-couple and other sources. According to this decomposition, a diagonalized tensor M consists of the following components:

$$M = M_{iso}+M_{dev} = M_{iso}+aM_{dc}+bM_{clvd} = M_{iso}+a\begin{pmatrix}-1 & 0 & 0\\ 0 & 0 & 0\\ 0 & 0 & 1\end{pmatrix}+b\begin{pmatrix}-1 & 0 & 0\\ 0 & -1 & 0\\ 0 & 0 & 2\end{pmatrix}$$

where M_{iso} is the isotropic part of the tensor and M_{dev} the deviatoric part, which is decomposed into a double-couple component M_{dc} and a *compensated linear vector dipole* (CLVD) component M_{clvd}, with a and b depending on the eigenvalues of M. While this is not the only valid moment tensor decomposition, it allows classification of moment tensors into seismic events that belong to the following groups:

- Highly isotropic: Force signatures caused by (artificial) explosions or implosions are isotropic.
- Mainly DC: Ideal faulting corresponds to slip along a single fault plane.
- Large CLVD components: Slipping along multiple faults, planar movement, or volcanic activities lead to high CLVD parts.

As presented in a later section, this classification allows predictions about moment tensor glyph shapes. During data analysis or processing, moment tensors are often represented by an approximating best double-couple, neglecting isotropic or CLVD contributions.

7.2.3. Moment Tensor Properties

In contrast to tensor data caused by man-made explosions, moment tensors derived from wave emissions of natural causes are usually traceless tensors, since these movements cause no changes in volume. Therefore, positive wave polarity or displacement along one direction is always accompanied by negative displacement along another direction, as indicated by opposing signs in the tensor eigenvalues. As a consequence, such moment tensors are indefinite:

$$\exists_{v,w} : \; v^T \cdot M \cdot v > 0 \;\wedge\; w^T \cdot M \cdot w < 0 \tag{7.2}$$

with $v, w \in \mathbb{R}^3$. This important property of the quadratic form $q_M(v) = v^T \cdot M \cdot v$ can be used directly for tensor visualization, as detailed in the next section.

$$\begin{aligned} M_0 &= \sqrt{\frac{1}{2}\sum_{ij} m_{ij}^2} \\ M_w &= \frac{2}{3} log_{10} M_0 - 10.7 \end{aligned} \tag{7.3}$$

As additional quantities derived from moment tensor data, the seismic moment M_0 and the resulting moment magnitude M_w defined by (7.3) provide means to measure the size of an earthquake.

7.3. Moment Tensor Glyphs

As described earlier, tensor glyphs are the preferred way to visualize well-defined properties of individual properties. Beachball glyphs as shown in Figure 7.1 (c) for a pure double-couple are the classic way to depict fault plane orientations of moment tensors in geoscience, where a circle is shaded according to wave polarity. Black indicates tensional, white compressional forces. Fault plane candidates can be perceived from the glyphs, as they are aligned with the borders between black/white quadrants. Positive (negative) definite tensors yield completely black (white) glyphs, while CLVD moment tensors lead to non-orthogonal shading of the glyphs. Most publications focus on the depiction of the double-couple component of a moment tensor to identify prevailing fault plane orientations. However, neither do these glyphs convey clear information about relative amplitudes of waves even when the shapes are scaled by M_0 or M_w to show earthquake magnitude, nor do they always provide a clear look at slip directions. We overcome these drawbacks by exploiting the fact that most moment tensors are indefinite and carry important information in their quadratic form. The glyphs developed in the following are capable of visualizing important tensor features such as local tensor definitness and slip geometry.

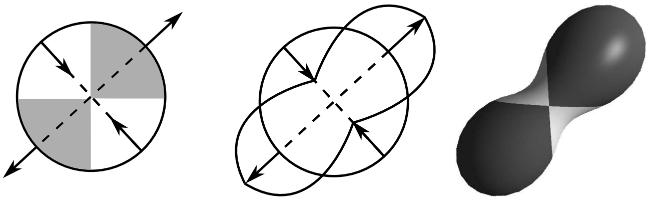


Figure 7.2.: Standard beachball and glyph shape offsets according to q_M. The shape of the resulting glyph conveys direction and magnitude information.

7.3.1. Polarity Glyphs

For positive definite diffusion tensors, several authors [ÖM03] have proposed to visualize M by extracting and rendering the implicit glyph $q_M(v) = c$ or $q_{M^{-1}}(v) = c$ for a constant $c \in \mathbb{R}^+$, thus depicting scaling and diffusion properties of the tensor. In contrast to these methods, we use the quadratic form q_M to modulate a basic glyph shape to visualize wave polarity and amplitude of indefinite tensors. The final glyph shape is desired to allow distinction between sources with different wave polarities, directions, absolute and relative amplitudes. Given an arbitrary unit vector in spherical coordinates $v(\theta, \phi) \in \mathbb{R}^3$, $q_M(v)$ represents the signed magnitude of the force couple in direction of v. For $v = e_i$, $i \in \{1, 2, 3\}$, the quadratic form evaluates to the corresponding eigenvalue. Motivated by these properties, vertices of an isotropic mesh can be modified to produce the desired glyphs, as seen in Figure 7.2:

1. Create a spherical mesh S with radius r
2. Compute $q_M(v/||v||)$ for every vertex $v \in S$
3. Displace mesh points v by $d_M(v) = c \cdot q_M(v/||v||) \cdot \frac{v}{||v||}$

Both r and c are determined by the desired scaling preferences described in the following. For a given set of moment tensors, seismic moments and therefore extremal values of q_M may vary strongly. The straightforward approach to yield consistent glyph scaling over a set of moment tensors is to choose $c = \frac{r}{max_M(|q_M|)}$ and fix r to a constant value over the data set. This way relative differences in glyph shape displacements d_M are equivalent to relative differences in seismic moments. Furthermore, glyphs shapes do not self intersect, as $||d_M(v)|| \leq r$ holds for all M. While this method accurately and consistently reflects force magnitudes by glyph displacements, glyphs with comparatively small wave amplitudes degenerate to sphere-like shapes as observable in Figure 7.3 (a), as earthquake magnitudes are reflected in glyph silhouettes directly. As a second method, we choose $c = \frac{r}{max_v(|q_M|)}$ with r proportional to M_w for every glyph individually. This

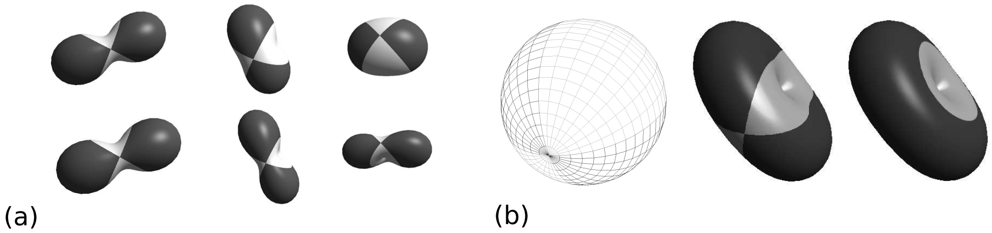


Figure 7.3.: (a) Glyphs scaled according to method one (top row) and method two (bottom row). Texturing shows orientation of the best double-couple. (b) Basic rotated mesh and glyph of a tensor with large CLVD component, showing double-couple and polarity shading.

results in glyphs that show accurate force and displacement ratios within a glyph and represent differences in earthquake magnitude between glyphs by uniform scaling. The advantage of latter method is that directions and polarity of maximal forces within a glyph are depicted more clearly. A drawback of this uniform scaling in three dimensions is, however the ambiguity of size and distance, which requires earthquake magnitude to be mapped to an additional visual parameter such as color.

7.3.1.1. Visualization

Following the shading convention of beachball glyphs, we make use of texturing to either highlight the best double-couple of a glyph by appropriate shading of the quadrants, or map positive and negative polarity to different colors. Figure 7.3 (b) shows examples of both shading types. Polarity based shading allows easy identification of tensors with large isotropic or CLVD components. We note that spherical meshes are rotated prior to vertex displacement, to ensure that high resolution poles of the discretized mesh coincide with the direction of e_2. This allows accurate texturing of the region where the four quadrants meet. All glyphs are shaded and textured using pixel shaders to avoid visual artifacts resulting from low resolution meshes and aid in conveying glyph shape properties such as curvature. When compared to classic beachball renderings, polarity glyphs are able to not only represent fault plane orientation but CLVD and isotropy fractions and absolute displacement magnitudes as well.

7.3.2. Slip Geometry

As noted by Haller [Hal01, Hal05] in the context of flow vortices, indefinite matrices describe a partition of local space into sections with attracting and repelling flow behavior. The polarity glyphs introduced in the last section encode this attracting and repelling behavior in the context of wave propagation and force orientation by shading surface areas accordingly. To convey the full shape of this flow

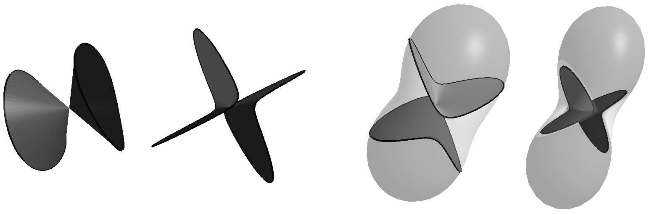


Figure 7.4.: Slip geometry glyphs for high CLVD and high DC tensors (left) and geometry glyphs rendered together with polarity glyphs (right).

separation or fault geometry, we propose a method to extract slip geometry G_S of a tensor directly. For a given indefinite tensor, the quadratic form q_M defines a unique scalar field on the unit-sphere, with a zero set $\{v|q_M(v) = 0\}$ corresponding to the closed boundary between regions of inflow and outflow or opposing force directions. In moment tensor data, this boundary indicates slip geometry. To extract this geometry, we discretize the unit-sphere uniformly along its spherical coordinates θ and ϕ. The resulting grid cells $[i\Delta\theta, (i+1)\Delta\theta] \times [j\Delta\phi, (j+1)\Delta\phi]$ are quads in spherical coordinate space and thus qualify for standard isocontour extraction techniques. For the isovalue of $q_M(v) = 0$, performing the standard Marching Quads algorithm on these cells yields a connected set of lines on the unit-sphere. Slip geometry triangles are created by connecting these line segments with the origin of the unit sphere. These triangles are a valid representation of G_S, as $q_M(w) = 0$ holds for all w with $v^T \cdot \frac{w}{||w||} = 1$.
As isotropic tensors do not contain a distinct direction of slip, we only apply this method to anisotropic tensors, as common in real world data sets.

7.3.2.1. Visualization

Visualizations of G_S for different moment tensors are shown in Figure 7.4. Highlighted rendering of glyph boundaries helps distinguish different slip geometry types. Slip and polarity glyph types complement each other when combined. Pure double-couple tensors produce two perpendicular flat sphere segments as expected, whereas CLVD tensors produce more complex geometries due to different contributions of a number of double-couple fault planes to the final shape of the glyph. In contrast to beachball or polarity glyphs, this extracted geometry gives a clear look at interior displacement discontinuities. Highest information density is reached when combined with transparent polarity glyphs.

7.4. Clustering and Averaging

The presented glyphs facilitate a local visualization and analysis of scattered moment tensor fields. As straight-forward interpolation of scattered moment tensors is not feasible due to the loss of important tensor properties when mixing different source types, tensor grouping or averaging has to rely on additional clustering steps. For an insightful visualization of the complete tensor field, several reasons motivate the averaging and clustering of multiple tensors into subsets. A first reason is given by visual occlusion that frequently appears in regions with a high density of seismic activities being represented by large and dense groups of moment tensor glyphs. A second reason is the fact that accumulations of spatially close moment tensors commonly represent similar faulting mechanisms. Similar faulting mechanisms may be summarized into an abstract average seismic event, thus reducing information overload. Another reason for moment tensor clustering is the identification of locations and properties of frequently active seismic sources. In the following, we present a selection of moment tensor similarity measures and introduce a novel tensor clustering method along with an averaging procedure, allowing efficient computation and visualization of tensor clusters.

7.4.1. Similarity Measures

For diffusion tensors, a number of metrics were proposed over the last decades [PSB10]. Among others, Kagan[Kag91] and Willemann[Wil93] present similarity measures for moment tensors. While Kagan defines similarity between moment tensors as the minimum angle needed to transform one double-couple into another by rotation, Willemann uses the inner tensor product to define similarity on the full tensor rather than the best double-couple. In the following we summarize both approaches and introduce modifications to allow source-type clustering.

7.4.1.1. Double-Couple Rotation

The double-couple orientation of a moment tensor is fully described by the eigenvector system of the tensor. For a fixed ordering of eigenvectors such as $e_x, e_y, e_z \simeq e_1, e_2, e_3$, this orthogonal system can be converted into a unit-quaternion[BH87] representation $Q = (cos(\alpha/2), sin(\alpha/2) \cdot q) = (q_0, q_1, q_2, q_3)$ by:

$$\begin{pmatrix} e_1 & e_2 & e_3 \end{pmatrix} = \begin{pmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2(q_1q_2 - q_0q_3) & 2(q_1q_3 + q_0q_2) \\ 2(q_1q_2 + q_0q_3) & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2(q_2q_3 - q_0q_1) \\ 2(q_1q_3 - q_0q_2) & 2(q_2q_3 + q_0q_1) & q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{pmatrix}$$

This right-handed (left-handed) coordinate system is not unique but results in 4 different quaternions depending on choice of eigenvector signs, from which one commonly chooses the solution with minimal rotation angle to the canonical coordinate frame. For quaternions Q_M, Q_N of two moment tensors, the minimal angle of rotation between double-couples of M and N corresponds to

$$\alpha_{min}(M, N) = 2 \cdot cos^{-1}(max_q((Q_M \cdot q \cdot Q_N^{-1})_0)) \tag{7.4}$$

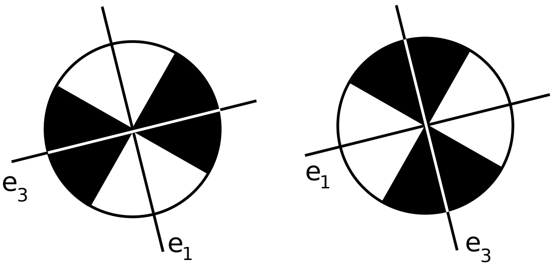


Figure 7.5.: Two moment tensors describing different earthquakes with identical fault geometry. $\alpha_{min} = \Pi$, $\hat{\alpha}_{min} = 0$, $S = \Pi$ and $\hat{S} = 0$.

where $q \in \{(1,0,0,0),(0,1,0,0),(0,0,1,0),(0,0,0,1)\}$, and $(.)_0$ corresponds to the scalar component q_0 of a quaternion. This distance function α_{min} can be used to define a similarity measure on double-couples. Thus, when neglecting CLVD and isotropic components of moment tensors, this defines a metric on the approximating double-couples of arbitrary moment tensors (with the exception of isotropic tensors).

We modify this distance function to not only serve as a similarity measure on seismic wave propagation of earthquakes, but as well on fault geometry, see Figure 7.5. After computing α_{min}, we repeat computation of α_{min} with swapped directions e_1 and e_3 in the coordinate system of one of the two moment tensors. The overall minimal angle $\hat{\alpha}_{min}$ of both computations is equal to the minimal angle needed to rotate one fault plane system into the other, as illustrated in Figure 7.5. While slipping directions of moment tensors that are similar to each other with respect to $\hat{\alpha}_{min}$ may have opposing directions, this measure identifies moment tensors based on similarity of fault geometry.

7.4.1.2. Full Tensor Similarity

In contrast to the double-couple similarity measure described above, this distance function is not restricted to approximating double-couples, but operates on the full moment tensor data. For two second order square tensors M,N with $\sum_i \lambda_i^2 = 2$, the inner tensor product

$$M \cdot N = \sum_{ij} m_{ij} \cdot n_{ij} \tag{7.5}$$

is a natural measure of tensor similarity. Values of the product are limited to the interval $[-2, 2]$ with $M \cdot M = 2$ and $M \cdot -M = -2$. A function

$$S(M,N) = cos^{-1}\left(\frac{M \cdot N}{2}\right) \tag{7.6}$$

defines a distance between arbitrary normalized moment tensors M and N. S is maximal for opposing moment tensor pairs, i.e. for moment tensor pairs with identical orientation in the eigenvector system and opposite eigenvalue signs. We again propose a modified similarity measure by identifying moment tensors with identical fault orientations. For this matter, we replace $M \cdot N$ by $|M \cdot N|$ in the computation of S to yield a new distance measure $\hat{S}$ with similar properties as $\hat{\alpha}_{min}$ as indicated in Figure 7.5. In contrast to $\hat{\alpha}_{min}$, this distance function is able to operate on purely isotropic tensors as well. Aside from this fact, choice of the similarity measure is based on whether to include CLVD components into the clustering algorithm or not. If the data is suspected to contain a lot of noise, causing pure double-couple sources to include incorrect CLVD components, the first measure might therefore be preferred over the latter.

In the following the four presented distance functions are used to realize interactive and automatic tensor clustering. It is important to note that both presented distance measures ignore the magnitude of the seismic moment either by using the eigenvector system directly, or by operating on normalized tensors.

7.4.2. Clustering

Clustering commonly serves to group and identify similarities in arbitrary data sets, where its data reduction and information densification properties make it a prime candidate to be applied to cluttered tensor glyph visualizations. In the context of this work, tensor clusters have to fulfill two properties. Firstly, tensors in a cluster have to be significantly similar to each other, secondly clusters should be constrained to a local neighborhood to avoid grouping of spatially separate tensors. A clustering of objects based on a similarity measure fulfilling the first property is given by the Quality-Threshold Clustering algorithm [HKY99]. To satisfy the second property, we propose a Distance-Quality-Threshold Clustering algorithm utilizing a minimum spanning tree representation of each cluster.

7.4.2.1. Distance-Quality-Threshold Clustering

Our DQT-Clustering algorithm is a modified version of the classic QT-Clustering method that additionally imposes euclidean distance constraints on the final clusters. Input parameters of the clustering algorithm are a distance function d such as α_{min}, a threshold ϵ_d that specifies the maximal allowed distance in similarity between two members of a cluster and a euclidean distance threshold ϵ_e defining the maximal euclidean distance between nearest neighbors of a cluster. The clustering algorithm takes the following form:

1. Let U denote the set of unclustered tensors and C_C the set of clusters.

2. Choose a tensor $M \in U$, starting a new candidate cluster $C_M = \{M\}$.

3. Add $argmin_{P \in U/C_M}(max_{N \in C_M}(d(P,N)))$ to C_M, until $max_{N \in C_M}(d(P,N)) > \epsilon_d$ for all $P \in U/C_M$.
4. Build a euclidean minimum spanning tree for C_M.
5. Remove all parts of C_M that are connected to M by an edge larger than ϵ_e.
6. Repeat from 2 with new M, until all tensors in U have generated a candidate cluster.
7. Pick the largest candidate cluster and add it to C_C. Remove its members from U. Clear all candidate clusters.
8. Repeat from 2, until $U = \emptyset$.

When execution of the algorithm has finished, C_C contains non-overlapping tensor clusters. With exception of steps four and five, this clustering algorithms is identical to the standard QT method. In step three the point P that minimizes the maximal distance to all members of the candidate cluster C_M is added to the set, if $max_{N \in C_M}(d(P,N)) > \epsilon_d$. This guarantees complete linkage clustering, i.e. the maximal distance with respect to d between two arbitrary members of a cluster is limited by ϵ_d. To minimize computational complexity, we pre-compute d for pairs of moment tensors during loading of the data set, thus ensuring reduced computation times during clustering.
While these steps yield clustering in the space of tensor similarity, they do not impose constraints on euclidean space, resulting in clusters that are distributed over large parts of the scattered tensor field. To enforce spatial locality of moment tensor clusters, we introduce a cluster splitting method based on minimum spanning trees. For a set of objects in euclidean space, a euclidean minimum spanning tree corresponds to a tree representation of the set with a minimal total of edge lengths among all trees that can be constructed from that set. We construct the minimum spanning tree for a candidate cluster using Prim's algorithm. By removing objects from the cluster that are connected to M by edges exceeding the pre-defined threshold ϵ_e, cluster locality is guaranteed, as illustrated by Figure 7.6 (a). Connectivity information gathered in this tree-building process is saved for visualization purposes.

7.4.3. Tensor Averaging

Clusters obtained by the described method facilitate local tensor averaging while excluding tensors of strongly dissimilar types, leading to an accurate representation of a closed set of moment tensors. Given a cluster of moment tensors M_i, we compute a representative tensor M_a by averaging of eigenvector directions and eigenvalues of cluster members. For M_i, $i \in \{1,..,n\}$ with ordered eigenvectors e_1^i, e_2^i, e_3^i, eigenvectors of M_a evaluate to

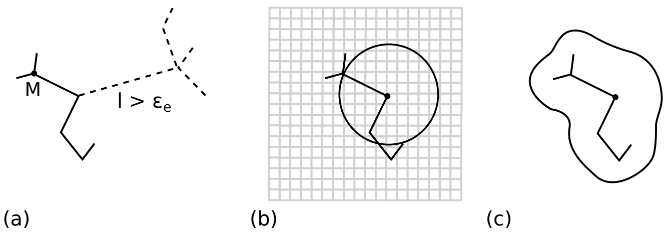


Figure 7.6.: (a) Minimum spanning tree for a set of points in $\mathbb{R}^2$. Parts of the tree are removed, as connecting edges exceed ϵ_e. (b) Metaball radius for a selected node with underlying extended cluster bounding box. (c) Illustration of final cluster outline.

$$(e_1^a, e_2^a, e_3^a) = \left(\frac{\sum_i e_1^i}{||\sum_i e_1^i||}, \frac{\sum_i e_2^i}{||\sum_i e_2^i||}, \frac{\sum_i e_3^i}{||\sum_i e_3^i||} \right) \tag{7.7}$$

with individual eigenvectors e^i oriented according to a chosen right-hand coordinate system implied by (e_1^0, e_2^0, e_3^0). Eigenvectors of isotropic sources are omitted; corresponding eigenvalues are averaged similarly. If $\hat{\alpha}_{min}$ or $\hat{S}$ was used during clustering, e_1^i and e_3^i may be swapped during the averaging process to correspond to the directions of e_1^0 and e_3^0. From this eigenvector system, the average pseudo moment tensor is reconstructed by solving

$$M_a \cdot \begin{pmatrix} e_1^a & e_2^a & e_3^a \end{pmatrix} = \begin{pmatrix} \lambda_1^a \cdot e_1^a & \lambda_2^a \cdot e_2^a & \lambda_3^a \cdot e_3^a \end{pmatrix}. \tag{7.8}$$

Glyphs of these average tensors show an accurate representation of average fault plane directions and polarity magnitudes. For source type clustering we limit display of average tensors to slip geometry glyphs, as wave polarity is neglected during that type of clustering. The representative glyph of a cluster is placed at the position of an edge of the minimum spanning tree that is closest to the centroid of the cluster's bounding box.

7.4.3.1. Cluster Visualization

While classic ways of visualizing object clusters by distinct object colors or shapes allow visual distinction of clusters, they often fail to convey a homogeneous representation of a cluster and rely on the visual presence of cluster members. Cluster outlining techniques avoid these problems. We propose a cluster outlining technique based on implicit metaball [Bli82, LAG01] shapes to allow consistent cluster visualization in the absence of cluster members.

Based on the minimum spanning tree skeleton obtained during clustering, information about moment tensor connectivity in a cluster is available and can be used in the process of visualization. For cluster outline generation we center an energy function

$$f(p_i, p) = max\left(0, \left(1 - \frac{||p - p_i||^2}{r_i^2}\right)^3\right) \tag{7.9}$$

at every member tensor position p_i with a radius r_i corresponding to the maximal length of adjacent edges in the minimum spanning tree, as seen in Figure 7.6 (b). This energy function has similar properties as the commonly used Gaussian function, but has an important advantage in evaluation speed. The accumulation of energy functions in a cluster yields a three-dimensional scalar field, that serves as basis for isovolume extraction. For discretization purposes, the cluster is enclosed by an axis aligned bounding box that has been enlarged by $max_i(r_i)$ into the canonical coordinate axes and sampled by a uniform grid, see Figure 7.6 (b). Scalar values $s(.)$ at grid node positions g_j are evaluated as $s(g_j) = \sum_i f(p_i, g_j)$. We use the standard Marching Cubes technique to extract a triangulated isovolume for a value of 0.6 in the resulting uniformly sampled scalar field. Hereby, empirical choice of isovalue and choice of the r_i produce a connected outline of the cluster.
A rendering of this isovolume geometry may be used for direct cluster visualization. However, we restrict rendering of these isovolumes to their view dependent silhouettes to provide a clear look at the cluster contents. To achieve this, we make use of the OpenGL stencil buffer to mask out irrelevant parts of the isovolume:

1. Choose a cluster C with $|C| > 1$.
2. Clear the stencil buffer bits to 1.
3. Draw the triangulated isovolume of C to the stencil buffer only, setting covered bits to 0.
4. Draw an enlarged version of the isovolume with shading and coloring to pixels where the stencil buffer is 1.
5. Repeat from 1 until all desired clusters are drawn.

An enlarged version of the isovolume used during rendering in step 4 is obtained by displacing isovolume vertices along their normals. As a result of this rendering technique, a view dependent outline of the cluster is drawn, providing a clear view at inner parts of the cluster, while depicting the silhouette of the cluster. An illustration is given in Figure 7.6 (c). These outlines provide a visual clue about cluster dimensions, even if no cluster members are drawn.

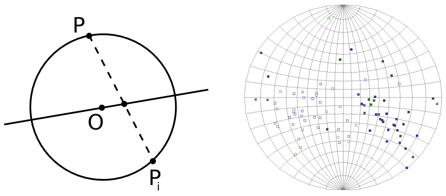


Figure 7.7.: Depiction of stereographic mapping (left) and an example of a mapping showing longitudinal and equatorial lines along with major and minor eigenvector directions of a set of moment tensors (right).

7.5. Interactive Visualization

In the previous sections, we have described methods to visualize and cluster scattered moment tensor fields in $\mathbb{R}^3$. This section details how to enhance visualization of such fields by the addition of an interactive projective rendering of the data set.

7.5.1. Stereonet Display

Lower hemisphere stereographic projections, or *stereonets* are angle preserving projective mappings that are frequently used in geoscience to analyze orientations of moment tensor data. A generalized stereographic projection for a projection point P on the unit-sphere maps points P_i on the unit-sphere surface to the intersection point of the line $g_i(t) = P + t(P - P_i)$ with the plane orthogonal to $(O - P)$ going through the origin. This mapping is undefined for P and is restricted to the hemisphere opposite to P in the following. An illustration of the mapping process along with a concrete example of a projected hemisphere are shown in Figure 7.7. To give a global impression of fault directions and moment tensor orientations, we extend the three-dimensional rendering context by a viewport showing such a stereonet type visualization. We render moment tensor orientations as points on the unit-sphere corresponding to minor and major eigenvector directions of the tensors. Major eigenvector directions are traditionally rendered as filled, minor eigenvector directions as empty dots. Due to orthogonality, the double-couple and fault plane directions of a moment tensor are uniquely defined by these two positions. As we detail in the results section, this view allows detailed cluster analysis of moment tensor orientations, while incapable of clearly depicting spatial information.

7.5.2. Interaction

To make full use of combined three-dimensional and projective display of a scattered moment tensor field, we provide both views with a list of interaction capabilities:

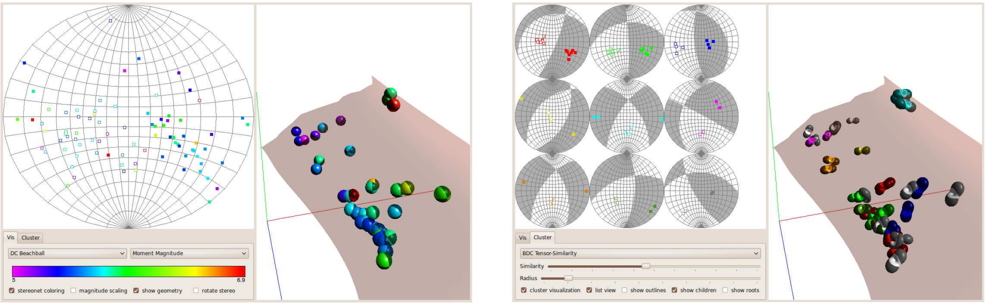
- Highlighting: When pointing at an eigenvector position in the stereographic view, both corresponding eigenvector directions are highlighted and connected to the sphere origin by line drawing while rendering a projected double-couple watermark in the background of the stereonet. The corresponding glyph in the three-dimensional view is highlighted by color and size changes to allow for easy matching of orientation and position and glyph display in $\mathbb{R}^3$.
- Selection: Users are able to use mouse based multiple tensor selection in both views to select, highlight or hide tensors. Affected glyphs and eigenvectors are highlighted, or hidden in both views.
- Parameter Selection: We provide the user with means to select the desired distance function for clustering, adjust ϵ_e, ϵ_d and change glyph type display as well as coloring (earthquake magnitude, CLVD fraction, isotropic fraction).
- Similarity Querying: Right click on a tensor M highlights all tensors N (glyphs and projected eigenvector directions) that satisfy $d(M,N) < \epsilon_e$ for the currently selected distance function and threshold.
- Clustering: Users can initiate clustering, toggle the display of individual cluster outlines, cluster members or representatives in both views.

All interaction methods aim at improving visual matching between the stereographic and the three-dimensional view of the data. Furthermore, they are designed to allow manual data exploration and analysis.

7.6. Results

In the following, we present results obtained by applying the methods described in this work to real world data sets. We visualize two data sets containing moment tensor data obtained during earthquakes in Chile and Tonga. Figure 7.8 gives an impression of standard beachball based moment tensor visualization together with a stereonet projection with fixed projection point and slab geometry display. As indicated by the color legend shown in the figure, visualization of scalar quantities is performed in HSV color space. The same figure shows a basic color based clustering visualization using polarity glyphs. Polarity glyphs facilitate easy identification of wave propagation directions and CLVD tensors, by conveying magnitudes and directions by tensor shapes. The alternative rendering of the stereonet as a list view, showing one stereonet for each cluster, allows uncluttered visualization of different clusters in projective space. Watermarking using average double-couple improves perception of fault directions in stereonet display. In general, the stereonet display gives an insight into the distribution of positive and negative wave propagation directions which is further aided by polarity glyph display. This view-based multi-field visualization facilitates independent but semantically linked analysis of eigenvector directions and tensor entities and follows a different principle than single-view multi-field visualization techniques. As opposed to latter methods, multi-view techniques reduce visual occlusion between different fields, but rely heavily on interaction to visualize local field relationships.

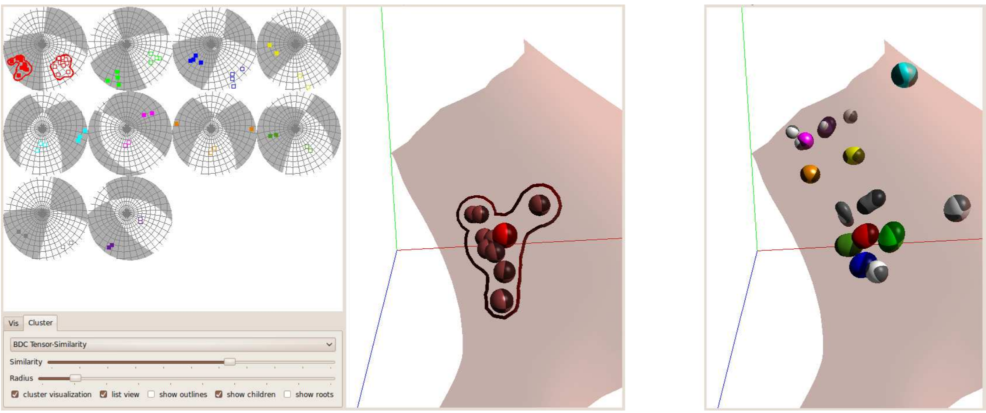
Figure 7.9 shows the benefit of information reduction by clustering. Clusters may be visualized by average tensors, outlines, cluster members or a combination of these. Cluster representatives are highlighted by color and size to be distinguishable from regular cluster members. The selected distance function (double-couple rotation) shows clear and correct clustering in both projective and three-dimensional space. Cluster outlines show information about spatial location and extension of frequent earthquake sources while not hiding individual earthquake sources.
Similarity querying as shown in Figure 7.10 is useful for quick identification of similar earthquake sources. Compared to clustering, it does not restrict its solution to local regions, but highlights tensors in all regions of the data set as made obvious by a single highlighted moment tensor at the further end of the data set. While display of all clustering data such as outlines, and cluster members, as seen in Figure 7.10, may lead to a cluttered view if no manual selection is performed, it can be used to gain a quick overview of the structure of a moment tensor data set. For cluster outlining in stereographic space, the minimum spanning tree is not created in three-dimensional euclidean space, but corresponds to a minimum spanning tree on the surface of the sphere, connecting eigenvector directions along geodesic lines of the sphere. This tree commonly leads to two cluster outlines per cluster, separating major and minor eigenvector directions, as seen in Figures 7.9 and 7.10.
Figure 7.11 shows a selection of visualization settings for a small moment tensor cluster including outline visualization. The depicted average tensor is scaled to be distinguishable from regular cluster members. Correct averaging results produce a well-aligned representative glyph. Correct cluster outlining is performed for arbitrary viewing angles. In contrast to the other presented results, which show the Chile data set, Figure 7.12 compares the standard beachball visualization technique applied to the Tonga data set with our approach, yielding a far less cluttered view of the fault data. It additionally shows visual feedback produced during orientation highlighting. In the presented data sets, clustering of moment tensors as well as geometry construction took less than a second on a PC with standard hardware (Intel Core 2 Duo @ 2 Ghz) and does not limit interactivity for small or medium sized data sets. For bigger data sets with a large number of tensors, clustering methods based on QT clustering need additional optimizations, which are outside of the scope of this work.



DC Beachball
Moment Magnitude
☑ stereonet coloring ☐ magnitude scaling ☑ show geometry ☐ rotate stereo

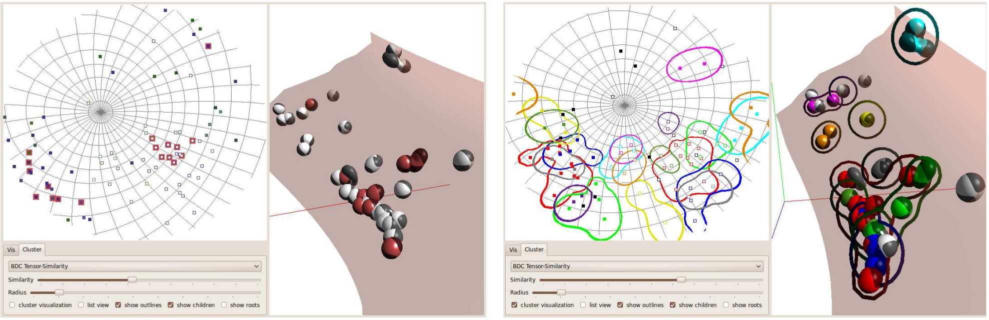
BDC Tensor-Similarity
Similarity
Radius
☑ cluster visualization ☑ list view ☐ show outlines ☑ show children ☐ show roots

Figure 7.8.: Standard display settings with fixed projection point and beachball glyphs colored according to moment magnitude (left). Visualization of clusters by polarity glyph coloring and a projective cluster-list showing double-couple orientation of cluster averages as watermarks (right).



BDC Tensor-Similarity
Similarity
Radius
☑ cluster visualization ☑ list view ☐ show outlines ☑ show children ☐ show roots

Figure 7.9.: View dependent stereonet projection along with cluster selection and outlining in $\mathbb{R}^3$ and stereonet view. The representative glyph is displayed along with cluster members (left). Replacing cluster members by their respective cluster averages leads to a simplified and decluttered view of the tensor field and shows locations of frequent and similar seismic activities (right).



BDC Tensor-Similarity
Similarity
Radius
☐ cluster visualization ☐ list view ☑ show outlines ☑ show children ☐ show roots

BDC Tensor-Similarity
Similarity
Radius
☑ cluster visualization ☐ list view ☑ show outlines ☑ show children ☐ show roots

Figure 7.10.: After user selection, tensors similar to a selected moment tensor are highlighted (left). Visualization of all identified clusters by outlining in projective and three-dimensional space (right).

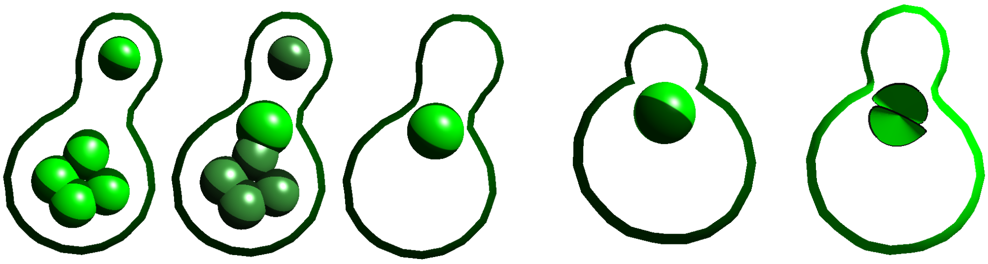


Figure 7.11.: A close up view of a cluster with outlining. Display of cluster member or the representative glyph can be toggled by the user (left). Correct outlining is preserved if changing the viewing angle (right). Display of slip geometry in a cluster allows a clearer look at geometrical segmentation in the form of fault planes and displacement discontinuities. Furthermore, the conic shape supports analysis of wave propagation directions and angles.

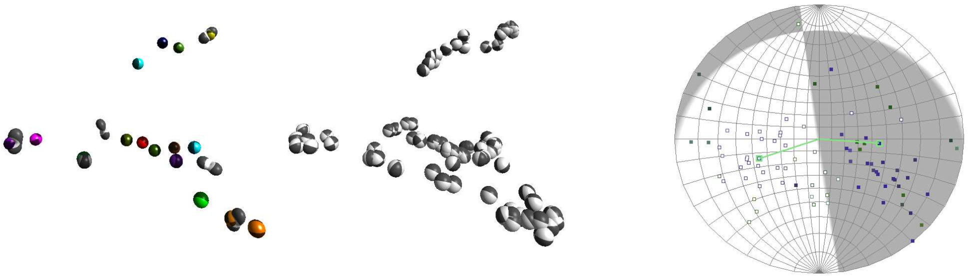


Figure 7.12.: Comparison of cluster averaging and polarity glyph display with classic unclustered beachball visualization (left). Moment tensor highlighting in stereonet view identifies related wave propagation directions and uses watermarking for fault plane orientation display (right).

7.7. Summary and Discussion

In this chapter we have introduced visualization techniques that aid the visual analysis of scattered moment tensor data. We combine novel glyph-based visualization techniques with stereonet based projective-renderings to obtain interactive visualizations in three-dimensions and projective space of eigenvector orientations. Glyph shapes and texturing facilitates the classification of individual earthquake sources by conveying prominent features of the bilinear form of the tensor. A new clustering technique operating in moment and euclidean space allows the efficient reduction of redundant information, leading to less cluttered and more structured visualizations. The created clusters can be used as guide to identify similar earthquake mechanisms or sources and introduce structure in the otherwise sparsely sampled scattered tensor field.

Various interaction techniques allow the direct connection of information from projective and three-dimensional space and are a crucial part of the multi-view visualization process. All presented methods lead to a higher information density than existing techniques and improve over the state of the art by conveying data properties such as definiteness that were previously hidden or hard to distinguish. Direction of future work may include the incorporation of further user preferences into the clustering process as well as enhanced display of additional geological information such as mantle geometry.

8. Conclusions

In this thesis we have developed novel feature extraction and segmentation techniques for scalar-, vector-, and tensor fields and emphasized their capability to perform multi-field visualization. The majority of the presented extraction techniques are based on scattered point sets and require the processing of derived tensorial quantities of the respective fields.

We have contributed a novel method for grid-less crease structure extraction in three-dimensional vector fields and have shown its use in complex-valued scalar fields from low frequency acoustics. Core crease structures are defined as maximal convexity ridges as revealed by eigen-decomposition of the Hessian of the scalar field. The extraction technique is based on curvature adaptive region growing and point convergence and was extended to extract non-manifold regions in the wave node structure. Multi-field visualization in this context focused on the display of amplitude and phase values obtained from the complex-valued scalar field as well as spatio-temporal display of multiple complex fields. Presented numerical results demonstrated the benefit of wave node analysis for acoustics design and engineering.

Furthermore, we investigated the effect and visualization of flow induced strain in unsteady flow fields. We extract and visualize strain tensors along integral flow lines by accumulating relative displacement given by the flow Jacobian. The resulting localized strain computations include strain notions as given by FTLE methods. The novel extraction and segmentation methods developed in this thesis demonstrated how the obtained strain information conveys relevant information in geophysical flow simulations. Resulting feature definitions allow multi-field visualization of the flow and strain tensor fields.

The effect of strain and displacement in a macroscopic context on integral flow feature extraction led to the definition of new adaptive streakline and streak surface generation techniques. We proposed distance and curvature based geometry refinement and discuss accuracy and artifact handling in three-dimensional surface generation. We used this adaptive geometry representation as boundary definition of flow segmentation, which facilitates the application of selective multi-field visualization.

New methods for the clustering and multi-field visualization of quantities related to geometric displacement, so called moment tensors from geophysical earthquake measurements were introduced, that facilitate semi-automatic interactive analysis of scattered tensor fields. Direct tensor visualization was performed by tensor glyphs that highlight indefiniteness of moment tensors by coloring and shape, whereas visualization of the derived scattered eigenvector field was carried out in orientation space in the form of stereonet displays.

With this thesis we have advanced the state of the art in visualization for scalar-, vector-, and tensor fields by the introduction of mesh-less and adaptive feature extraction techniques. Furthermore, we have developed new methods for the incorporation of derived fields such as strain and displacement into the visualization process, which allows for novel multi-field visualization techniques. We have emphasized the importance of strain and other tensorial measures for feature-based field analysis and developed techniques for single- and multi-view visualization of multiple fields. The presented numerical results demonstrate applicability and expressiveness of the introduced methods in areas such as sound simulation, industrial mixing, and geoscience.

For the analysis of strain in fields and scattered field approximation techniques, a number of challenges remain. The capability of strain to model convergence, divergence and hyperbolicity in general makes it a suitable tool for topological analysis of flow fields and adaptive modeling of flow features. In visualization, the incorporation of directional information in the visualization and analysis of FTLE fields is a promising topic of future work. We expect that a large group of existing visualization techniques can benefit from the incorporation of such derived and related field data.

A. Moving Least Squares

Least squares fitting allows polynomial approximation of discrete data. The main approximation technique used in this work, *Moving Least Squares* (MLS) [Lev98], is a generalization of *Least Squares Approximation.* In the following, we give a brief introduction into the mathematical background of MLS approximation and state the challenges resulting from the use of MLS as scattered data approximation technique.

A.1. Definition

Least squares fitting provides a method to compute functions that minimize the squared distance to a given set of n discrete data points (p_i, f_i). If a local approximation of the data is desired, the classic Least Squares scheme may be generalized to the *Weighted Least Squares* method by the introduction of a weighting function ω with local support, see [Nea04]. A polynomial function f with given degree that satisfies the weighted least squares condition at a point of evaluation $p = (x\ y\ z)^T$ is defined by (A.1).

$$min \left\{ \sum_{i}^{n} \omega(p, p_i) || f(p_i) - f_i ||^2 \right\} \tag{A.1}$$

The scheme obtained from moving the Weighted Least Squares over the domain of the data set to yield a continuous approximation of the field, is called *Moving Least Squares* approximation. For appropriately continuous ω, the global field reconstruction function is continuous as well.

A.2. Properties

This section aims at clarifying relevant reconstruction properties of MLS fitting such as accuracy and derivative computation. For a detailed review of properties of polynomial approximation and *Radial Basis Functions* (RBF), we refer to [Wen04].

A.2.1. Accuracy

Reconstruction properties of MLS depend heavily on the chosen weighting function. In the case of ω being a two-dimensional Gaussian function, changes in the variance parameter, shape or radius of the smoothing function have great impact on the output with respect to scale space and reconstruction detail. Common exponential weighting functions of the general form

$$\omega(p, p_i) = a \cdot e^{\frac{||p-p_i||^2}{r^2}} + b$$

for example, yield different results for a varying *smoothing length* r. Hereby, an increased smoothing length leads to less detailed but smoother reconstructions. Thus, the reconstructed functions are highly sensitive to changes in smoothing length, as too small radii may introduce noise or lead to singular systems, and too large radii blur important features of the field and reduce reconstruction accuracy. Especially in data sets with inhomogeneous particle densities, choice of the appropriate smoothing radius has an influence on the reconstructed function and is also a major factor contributing to increased computation times.

A.2.2. Complexity

One of the main reasons why grid-less approximation techniques are outperformed by grid-based methods is the computationally expensive gathering and weighting step of particle neighbors that fall within the support region of the weighting function. In the case of linear MLS with a polynomial of the general form $f(p) = c \cdot (1\ x\ y\ z)^T$, the following linear system of equations needs to be solved for c:

$$\left(\sum_i^n \omega(p, p_i) \begin{pmatrix} 1 & x_i & y_i & z_i \\ x_i & x_i^2 & x_i y_i & x_i z_i \\ y_i & x_i y_i & y_i^2 & y_i z_i \\ z_i & x_i z_i & y_i z_i & z_i^2 \end{pmatrix} \right) c = \sum_i^n \omega(p, p_i) \begin{pmatrix} 1 \\ x_i \\ y_i \\ z_i \end{pmatrix} f_i \qquad \text{(A.2)}$$

To avoid a singular system, this linear three-dimensional MLS needs at least 4 non-collinear points p_i in the neighborhood of p. We note that the symmetric matrices resulting from a product of base-vectors on the left side of (A.2) is independent of the point of evaluation and is completely determined by the position of a data point p_i. Following this observation, matrix creation can be relocated to a pre-processing step, pre-computing one such matrix for each data point, thus removing the expensive matrix creation step from approximation. Computational complexity of this LSE is drastically increased by degree or dimension elevation of domain or range of the data set. So does quadratic three-dimensional approximation already require solving a 10 × 10 system once per dimension of f_i. It is notable that MLS matrices of higher order include those of lower order. Advantages of MLS over other scattered data approximation techniques are

Derivative	Value
$\frac{\partial f}{\partial x}$	$= c_1 + 2 \cdot x \cdot c_3 + y \cdot c_4$
$\frac{\partial f}{\partial y}$	$= c_2 + x \cdot c_4 + 2 \cdot y \cdot c_5$
$\frac{\partial^2 f}{\partial^2 x}$	$= 2 \cdot c_3$
$\frac{\partial^2 f}{\partial^2 y}$	$= 2 \cdot c_4$
$\frac{\partial^2 f}{\partial x \partial y}$	$= c_4$

Table A.1.: Derivative lookup table for $f(p) = c \cdot (1 \quad x \quad y \quad x^2 \quad xy \quad y^2)^T$.

mathematical simplicity, and ease of control over polynomial degree and approximation error.

A.2.3. Derivatives

Polynomial approximation allows fast computation of function derivatives. A polynomial of degree n allows analytic differentiation of the approximated function up to the nth derivative. In practice, computation of these continuous derivatives consist of a simple table lookup for known dimension and function degree. An example of a lookup table is given in Table A.1 for two-dimensional quadratic approximation.

A.3. Challenges

In this work, we overcome a number of challenges of robust and efficient MLS-based field approximation as described in [OHBKH11]. In the following, we briefly describe the most relevant problems:

Fast neighbor identification For fast neighborhood identification during the weighting process in MLS approximation [MG91, IL05], we sort field points into an implicit uniform grid structure with cell sizes corresponding to the diameter of the support of the used weighting function. During field evaluation, only points located in the 9 or 27-neighborhood of the cell containing the evaluation point are used for construction of the LSE, thus ensuring locality during data evaluation.

Visibility querying Neighborhood identification is not fully captured by the radial function used for MLS approximation, as flow obstacles might separate points that fall within the same neighborhood circle [BKF+96]. To resolve this issue, the neighborhood test has to be accompanied by a visibility test. For this matter, we implemented a binary ray-boundary intersection test that is performed between the evaluation point and neighbor candidates during point neighborhood gathering.

Furthermore, the correct use of weighted approximation techniques requires considerable effort with respect to weighting function design and parameter choice to allow faithful reconstruction of the approximated function and is still topic of active research. In cases, where we use MLS, we either choose the weighting function that corresponds to the kernel used in the simulation or give additional motivation for the choice made.

B. Voxelization

Voxelization is a discretization or sampling technique, that converts a given object into a volumetric representation of sets of volume elements, so-called *voxels*. There are two different approaches to geometry voxelization. The first class of algorithms is based on three-dimensional polygon rasterization that is performed on the CPU, the second class uses current graphics hardware to obtain a voxel representation of input geometry.

B.1. GPU Voxelization

While consumer graphics hardware allows efficient rasterization for two-dimensional texturing, direct rendering to volumetric textures is currently not supported. Therefore, the common technique to obtain volume renderings of geometry is to slice the object into n view-aligned slices and create a layered 3d texture from this set of two-dimensional rasterizations [FC00]. Implementations on state-of-the-art consumer hardware is able to achieve new-interactive framerates for high-resolution voxel grids [ZCEP07]. Generally, these methods however require to render geometry at least n times and leads to problems when rendering faces that are parallel to the viewing direction [ED06]. Additionally, the number of available slices in z-direction is limited, especially if voxelization is not only desired to produce voxel masks, but is used to rasterize additional quantities such as normals or color.

B.2. CPU Voxelization

CPU based voxelization techniques are technically only limited by the amount of memory. In general, a combination of the following steps [KS87] is performed triangle-wise to voxelize a generic triangulated mesh:

- Determine axis-aligned plane p with maximal area of triangle projection.
- Rasterize projected triangle edges on p.
- Scan-line fill the triangle interior on the plane
- Transform 2D rasterization into 3D space by applying correct z-values.

Figure B.1 illustrates the steps of triangle voxelization. Steps 2-4 can be performed with fast integer arithmetic. In addition to pure voxelization, additional quantities given at triangle vertices such as color or triangle normals can be interpolated during edge rasterization and scan-line filling to provide intermediate values at all voxels. It is important to note that for triangle edges the resulting voxelization has to be independent of the chosen initial projection plane to ensure that voxelization of neighboring triangles does not contain holes.

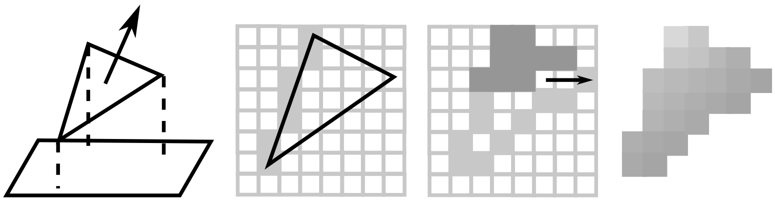


Figure B.1.: Steps of voxelization include projection, line rasterization, scan-line conversion, and depth propagation.

B.3. Volume Voxelization

If the goal of voxelization is to produce volume representations from mesh geometry, a filling step has to be performed after voxelization of the mesh boundary to mark the interior of the volume. For convex volumes with manifold boundary, this can be implemented by scan-line conversion of volume slices. A generalization of scan-line conversion to tetrahedra allows fast volume voxelization of tetrahedral volumes [KS87]. If the volume is concave or defined by multiple intersecting boundary meshes this requires additional effort [FC00, Lla07]. In the following we give details about volume voxelization based on flood filling. This concept of volume voxelization requires to voxelize volume boundaries in a first step , followed by a second step of recursive flood filling and classification of regions.

Let S be a set of boundary meshes in $\Omega \subseteq \mathbb{R}^3$. A boundary mesh f with the vector field f_n defining mesh normals implies a segmentation of space into regions f^0, f^+, f^- by

$$\begin{aligned}
f^0 &= \{x \in \Omega | \exists y, z : y \in N(x) \wedge (y = f(z) + f_n(z) \cdot \epsilon) \equiv (y = f(z) - f_n(z) \cdot \epsilon)\} \\
f^+ &= \{x \in \Omega | \exists y, z : y \in N(x) \wedge y = f(z) + f_n(z) \cdot \epsilon\} / f^0 \\
f^- &= \{x \in \Omega | \exists y, z : y \in N(x) \wedge y = f(z) - f_n(z) \cdot \epsilon\} / f^0
\end{aligned}$$

where $N(x)$ is the set of points that are connected to x without crossing one of the boundaries defined in S. An illustration of the different sets is given in Figure B.2.

In practice, regions of the texture are classified during flood filling, based on pixel adjacency to voxelized boundaries. This volume voxelization strategy requires the incorporation of normal information during the boundary voxelization process. If a newly filled voxel is adjacent to a voxelized boundary, the voxel is classified as interior or exterior based on these normal directions stored in boundary voxels. Classification of single voxels is propagated to all voxels of the filled volume to determine classification of the volume into f^0, f^+, or f^-.

After classification of all parts of the complete voxel volume, the volume interior is defined by logic operations on these predicates, as illustrated in Figure B.2 for $f^- \wedge g^- \wedge h^-$.

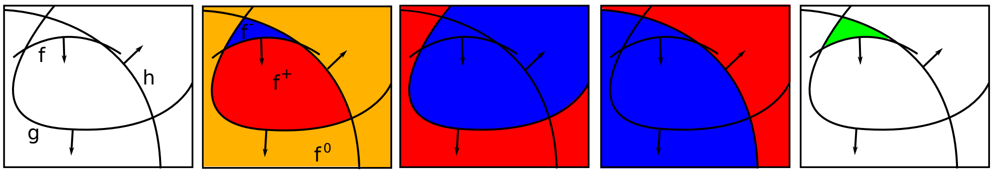


Figure B.2.: Illustration of region classification for voxel filling of a volume defined by $f^- \wedge g^- \wedge h^-$. Pixels that are defined as inside w.r.t. a mesh are marked in blue, outside pixels in red, ambiguous regions in orange. In this example, the intersection of all interior voxels is defined as the desired volume output.

Bibliography

[ABCO$^+$01] M. Alexa, J. Behr, D. Cohen-Or, S. Fleishmann, D. Levin, and C. T. Silva. Point set surfaces. In *Visualization '01*, pages 21–28, 2001.

[AD85] A. Agrawal and P. R. Dawson. A comparison of galerkin and streamline techniques for integrating strains from an eulerian flow field. *International Journal for Numerical Methods in Engineering*, 21:853–881, 1985.

[AG01] S. Akkouche and E. Galin. Adaptive implicit surface polygonization using marching triangles. *Computer Graphics Forum*, 20(2):67–80, June 2001.

[Ami02] I. Amidror. Scattered data interpolation methods for electronic imaging systems: a survey. *Journal of Electronic Imaging*, 11(2):157–176, 2002.

[AR80] K. Aki and P. G. Richards. *Quantitative seismology: Theory and methods*. Freeman San Francisco, 1980.

[Ban08] I. N. Bankman. *Handbook of Medical Image Processing and Analysis*. Academic Press, 2008.

[Bat67] G. K. Batchelor. *An Introduction to Fluid Dynamics*. Cambridge University Press, 1967.

[BDM$^+$05] M. Bertram, E. Deines, J. Mohring, J. Jegorovs, and H. Hagen. Phonon tracing for auralization and visualization of sound. In *IEEE Visualization 2005*, pages 151–158, Minneapolis, MN, USA, October 23–28 2005.

[BH87] K. Berthold and P. Horn. Closed-form solution of absolute orientation using unit quaternions. *Journal of the Optical Society of America*, 4(4):629–642, 1987.

[BHD$^+$94] M. Brill, H. Hagen, W. Djatschin, H. Rodrian, and S. V. Klimenko. Streamball techniques for flow visualization. In *Proc. of Vis '94*, pages 225–231, 1994.

[Bil08] M. I. Billen. Modeling the dynamics of subducting slabs. *Annual Reviews of Earth and Planetary Science*, 36:325–356, 2008.

[BJ02] P. J. Basser and D. K. Jones. Diffusion-tensor mri: theory, experimental design, and data analysis. In *NMR in Biomedicine*, volume 15, pages 456–467, 2002.

[BKF$^+$96] T. Belytschko, Y. Krongauz, M. Fleming, D. Organ, and W. K. S. Liu. Smoothing and accelerated computations in the element free galerkin method. *J. Comp. Appl. Math.*, 74(1-2):111–126, 1996.

[Bli82] J. F. Blinn. A generalization of algebraic surface drawing. *ACM Transactions on Graphics (TOG)*, 1(3):235–256, 1982.

[BML95] B. G. Becker, N. L. Max, and D. A. Lane. Unsteady flow volumes. In *Proceedings of the 6th conference on Visualization '95*, pages 329–335, 1995.

[Bot95] D. Botteldooren. Finite-difference time-domain simulation of low-frequency room acoustic problems. *J. Acoust. So. Amer.*, 98(6):3302–3308, December 1995.

[BPP$^+$00] P. J. Basser, S. Pajevic, C. Pierpaoli, J. Duda, and A. Aldroubi. In vivo fiber tractography using dt-mri data. In *Magnetic Resonance in Medicine*, volume 44, pages 625–632, 2000.

[Bra03] D. Braess. *Finite Elemente: Theorie, schnelle Löser und Anwendungen in der Elastizitätstheorie.* Springer, Berlin, 2003. 3. Auflage.

[Bre65] J. E. Bresenham. Algorithm for computer control of a digital plotter. *IBM Systems Journal*, 4(1):25–30, 1965.

[BT10] S. Barakat and X. Tricoche. An image-based approach to interactive crease extraction and rendering. *Procedia Computer Science*, 1(1):1703–1712, 2010. ICCS 2010.

[CB97] H. Chen and J. Bishop. Delaunay triangulation for curved surfaces. In *6th International Meshing Roundtable Proceedings*, pages 115–127, 1997.

[CBS07] C. P. Conrad, M. D. Behn, and P. G. Silver. Global mantle flow and the development of seismic anisotropy: Differences between the oceanic and continental upper mantle. *Journal of Geophysical Research*, 112, 2007.

[ČDFP05] L. Čomić, L. De Floriani, and L. Papaleo. Morse-smale decompositions for modeling terrain knowledge. In *Spatial Information Theory*, volume 3693 of *Lecture Notes in Computer Science*, pages 426–444. Springer Berlin / Heidelberg, 2005.

[Che93] L. P. Chew. Guaranteed-quality mesh generation for curved surfaces. In *9th Symposium on Computational Geometry*, pages 274–280, 1993.

[CK90] J. R. Cash and Alan H. Karp. A variable order runge-kutta method for initial value problems with rapidly varying right-hand sides. *ACM Trans. Math. Softw.*, 16:201–222, 1990.

[CKRW08] N. Cuntz, A. Kolb, Strzodka R, and D. Weiskopf. Particle level set advection for the interactive visualization of unsteady 3D flow. volume 27, pages 719–726, 2008.

[CL93] B. Cabral and L. C. Leedom. Imaging vector fields using line integral convolution. In *SIGGRAPH '93: Proceedings of the 20th annual conference on Computer graphics and interactive techniques*, pages 263–270, 1993.

[CM93] R. A. Crawfis and N. Max. Texture splats for 3d scalar and vector field visualization. In *Proceedings of the 4th conference on Visualization '93*, pages 261–266, 1993.

[CPK09] N. Cuntz, A. Pritzkau, and A Kolb. Time-adaptive lines for the interactive visualization of unsteady flow data sets. *Computer Graphics Forum*, 28(8):2165–2175, 2009.

[dAJ04] B. R. de Araujo and J. A. P. Jorge. Curvature dependent polygonization of implicit surfaces. In *Proceedings of the Computer Graphics and Image Processing*, pages 266–373, 2004.

[Dam99] J. Damon. Properties of ridges and cores for two-dimensional images. *J. Math. Imaging Vis.*, 10(2):163–174, 1999.

[DBM$^+$06] E. Deines, M. Bertram, J. Mohring, J. Jegorovs, F. Michel, H. Hagen, and G.M. Nielson. Comparative visualization for wave-based and geometric acoustics. In *IEEE Visualization 2006*, pages 1173–1179, Baltimore, Maryland, USA, October 29 – November 3 2006.

[DCH88] R. A. Drebin, L. Carpenter, and P. Hanrahan. Volume rendering. *SIGGRAPH Comput. Graph.*, 22(4):65–74, 1988.

[Dei08] E. Deines. *Acoustic Simulation and Visualization Algorithms.* PhD thesis, University of Kaiserslautern, Kaiserslautern, Germany, 2008.

[DH92] T. Delmarcelle and L. Hesselink. Visualization of second-order tensor fields and matrix data. In Arie E. Kaufman and Gregory M. Nielson, editors, *IEEE Visualization*, pages 316–323. IEEE Computer Society, 1992.

[DH93] T. Delmarcelle and L. Hesselink. Visualizing second-order tensor fields with hyperstreamlines. *IEEE Comput. Graph. Appl.*, 13(4):25–33, 1993.

[DH94] T. Delmarcelle and L. Hesselink. The topology of symmetric, second-order tensor fields. In *VIS '94: Proceedings of the conference on Visualization '94*, pages 140–147, 1994.

[dLvW93] W. C. de Leeuw and J. J. van Wijk. A probe for local flow field visualization. In *VIS '93: Proceedings of the 4th conference on Visualization '93*, pages 39–45, 1993.

[DZM07] R. Dyer, H. Zhang, and T. Möller. Delaunay mesh construction. In *Eurographics Symposium on Geometry Processing*, pages 273–282, 2007.

[Ebe96] D. Eberly. *Ridges in Image and Data Analysis.* Kluwer Academic Publishers, Dordrecht, The Netherlands, 1996.

[ED06] E. Eisemann and X. Décoret. Fast scene voxelization and applications. In *ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*, pages 71–78. ACM SIGGRAPH, 2006.

[Ede01] H. Edelsbrunner. *Geometry and topology for mesh generation.* Cambridge University Press, 2001.

[EHK+04] K. Engel, M. Hadwiger, J. M. Kniss, A. E. Lefohn, C. R. Salama, and D. Weiskopf. Real-time volume graphics. In *SIGGRAPH '04: ACM SIGGRAPH 2004 Course Notes*, New York, NY, USA, 2004. ACM.

[Eve01] C. Everitt. Interactive order-independent transparency. In *Technical Report. NVIDIA Corporation.*, 2001.

[EYD02] G. Erlebacher, D. A. Yuen, and F. Dubuffet. Case study: visualization and analysis of high rayleigh number — 3d convection in the earth's mantle. In *VIS '02: Proc. of the Conference on Visualization '02*, pages 493–496, 2002.

[FC00] S. Fang and H. Chen. Hardware accelerated voxelization. *Computers & Graphics*, 24(3):433 – 442, 2000.

[FCE+98] T. A. Funkhouser, I. Carlbom, G. Elko, G. P. M. Sondhi, and J. West. A beam tracing approach to acoustic modeling for interactive virtual environments. In *Computer Graphics (SIGGRAPH 98)*, pages 21–32, Orlando, FL, July 1998.

[FKMP97] J. Furst, R. S. Keller, J. Miller, and S. M. Pizer. Image loci are ridges in geometric spaces. In *SCALE-SPACE '97: Proceedings of the First International Conference on Scale-Space Theory in Computer Vision*, pages 176–187, 1997.

[FP98] J. Furst and S. M. Pizer. Marching optimal-parameter ridges: An algorithm to extract shape loci in 3d images. In *MICCAI '98: Proc. of the First Int. Conf. on Medical Image Computing and Computer-Assisted Intervention*, pages 780–787, London, UK, 1998. Springer-Verlag.

[GGTH07] C. Garth, F. Gerhardt, X. Tricoche, and H. Hagen. Efficient computation and visualization of coherent structures in fluid flow applications. *IEEE Transactions on Visualization and Computer Graphics*, 13:1464–1471, November 2007.

[Gin09] J.-M. Ginoux. *Differential geometry applied to dynamical systems.* World Scientific Publishing, 2009.

[GJL+09] E. Grundy, M. W. Jones, R. S. Laramee, R. P. Wilson, and E. L. C. Shepard. Visualisation of sensor data from animal movement. *Comput. Graph. Forum*, 28(3):815–822, 2009.

[GKS00] M. Gopi, S. Krishnan, and C. T. Silva. Surface reconstruction based on lower dimensional localized delaunay triangulation. In *Computer Graphics Forum 19*, 2000.

[GKT+08] C. Garth, H. Krishnan, X. Tricoche, T. Bobach, and K. I. Joy. Generation of accurate integral surfaces in time-dependent vector fields. *IEEE TVCG*, 6(14):1404–1411, 2008.

[GWT+08] C. Garth, A. Wiebel, X. Tricoche, K. I. Joy, and G. Scheuermann. Lagrangian visualization of flow-embedded surface structures. *Computer Graphics Forum*, 27(3):1007–1014, May 2008.

[Hab90] R. B. Haber. Visualization techniques for engineering mechanics. *Comput. Syst. Educ.*, 1(1):37–50, 1990.

[Hal01] G. Haller. Lagrangian structures and the rate of strain in a partition of two-dimensional turbulence. *Physics of Fluids*, 13(11):3365–3385, 2001.

[Hal05] G. Haller. An objective definition of a vortex. *Journal of Fluid Mechanics*, 525:1–26, 2005.

[HBH03] M. Hadwiger, C. Berger, and H. Hauser. High-quality two-level volume rendering of segmented data sets on consumer graphics hardware. In *VIS '03: Proceedings of the 14th IEEE Visualization 2003 (VIS'03)*, pages 301–308, 2003.

[HFH$^+$04] I. Hotz, L. Feng, H. Hagen, B. Hamann, K. I. Joy, and B. Jeremic. Physically based methods for tensor field visualization. In *VIS '04: Proceedings of the conference on Visualization '04*, pages 123–130, 2004.

[HGH$^+$10] M. Hummel, C. Garth, B. Hamann, H. Hagen, and K. I. Joy. Iris: Illustrative rendering for integral surfaces. *IEEE Transactions on Visualization and Computer Graphics*, 16:1319–1328, 2010.

[HH91] J. L. Helman and L. Hesselink. Visualizing vector field topology in fluid flows. *IEEE Comput. Graph. Appl.*, 11(3):36–46, 1991.

[HJYW03] Y. M. A. Hashash, I. John, C. Yao, and D. C. Wotring. Glyph and hyperstreamline representation of stress and strain tensors and material constitutive response. *Internat. Journal for Numerical and Analytical Methods in Geomechanics*, 27(7):603–626, 2003.

[HKY99] L. J. Heyer, S. Kruglyak, and S. Yooseph. Exploring expression data: identification and analysis of coexpressed genes. *Genome research*, 9(11):1106–1115, 1999.

[HS81] B. K. P. Horn and B. G. Schunck. Determining optical flow. *Artificial intelligence*, 17(1-3):185–203, 1981.

[HSH07] M. Hlawitschka, G. Scheuermann, and B. Hamann. Interactive glyph placement for tensor fields. In *Proc. of Third Internat. Sym. on Visual Computing*, pages 331–340. Springer-Verlag, 2007.

[HSIW96] A. Hilton, A. J. Stoddart, J. Illingworth, and T. Windeatt. Marching triangles: range image fusion for complex object modelling. In *Image Processing, 1996. Proceedings., International Conference on*, volume 1, pages 381–384 vol.2, Sep 1996.

[HSNHH10] I. Hotz, J. Sreevalsan-Nair, H. Hagen, and B. Hamann. Tensor field reconstruction based on eigenvector and eigenvalue interpolation. In Hans Hagen, editor, *Scientific Visualization: Advanced Concepts*, volume 1 of *Dagstuhl Follow-Ups*, pages 110–123. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany, 2010.

[Hul92] J. P. M. Hultquist. Constructing stream surfaces in steady 3d vector fields. In *VIS '92: Proceedings of the 3rd conference on Visualization '92*, pages 171–178, 1992.

[HY00] G. Haller and G. Yuan. Lagrangian coherent structures and mixing in two-dimensional turbulence. *Phys. D*, 147:352–370, 2000.

[IG97] V. Interrante and C. Grosch. Strategies for effectively visualizing 3d flow with volume lic. In *VIS '97: Proceedings of the 8th conference on Visualization '97*, pages 421–ff., 1997.

[Ihl98] F. Ihlenburg. *Finite element analysis of acoustic scattering.* Springer, New York, 1998.

[IL05] A. Iske and J. Levesley. Multilevel scattered data approximation by adaptive domain decomposition. *Numerical Algorithms*, 39:187–198, 2005.

[ITK05] The itk software guide, Nov. 2005.

[JB10] M. Jadamec and M. I. Billen. Reconciling surface plate motions with rapid 3d mantle flow around a slab edge. *Nature*, In Press, 2010.

[Kag91] Y. Y. Kagan. 3-D rotation of double-couple earthquake sources. *Geophysical Journal International*, 106(3):709–716, 1991.

[KC00] D. Kincaid and W. Cheney. *Numerical Analysis: Mathematics of Scientific Computing.* American Mathematical Society, 2000.

[KESW09] G. Kindlmann, R. S. J. Estepar, S. M. Smith, and C.-F. Westin. Sampling and visualizing creases with scale-space particles. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):1415–1424, 2009.

[KFW98] S. Khoury, A. Freed, and D. Wessel. Volumetric visualization of acoustic fields in cnmat's sound spatialization theatre. In *Visualization '98*, pages 439–442 & 562. IEEE, 1998.

[KGJ09] H. Krishnan, C. Garth, and K. Joy. Time and streak surfaces for flow visualization in large time-varying data sets. *IEEE Transactions on Visualization and Computer Graphics*, 15:1267–1274, 2009.

[KHL99] D. N. Kenwright, C. Henze, and C. Levit. Feature extraction of separation and attachment lines. *IEEE TVCG*, 5(2):135–144, 1999.

[Kin04] G. Kindlmann. Superquadric tensor glyphs. In *Proceedings of IEEE TVCG/EG Symposium on Visualization 2004*, pages 147–154, May 2004.

[KJKS08] S. Karato, H. Jung, I. Katayama, and P. Skemer. Geodynamic significance of seismic anisotropy of the upper mantle: New insights from laboratory studies. *Annual Review Earth and Planetary Science*, 36:59–95, 2008.

[KK06] S.-K. Kim and C.-H. Kim. Finding ridges and valleys in a discrete surface using a modified mls approximation. *Computer-Aided Design*, 38(2):173–180, 2006.

[KKKW05] J. Kruger, P. Kipfer, P. Kondratieva, and R. Westermann. A particle system for interactive visualization of 3d flows. *IEEE Transactions on Visualization and Computer Graphics*, 11(6):744–756, 2005.

[Koy97] J. Koyama. *The Complex Faulting Process of Earthquakes.* Kluwer Academic Publishers, Dordrecht, The Netherlands, 1997.

[KPH$^+$09] J. Kasten, C. Petz, I. Hotz, B. R. Noack, and H.-C. Hege. Localized finite-time lyapunov exponent for unsteady flow analysis. In Marcus Magnor, Bodo Rosenhahn, and Holger Theisel, editors, *Vision Modeling and Visualization*, volume 1, pages 265–274. Universität Magdeburg, Inst. f. Simulation u. Graph., 2009.

[KPI$^+$03] J. Kniss, S. Premoze, M. Ikits, A. Lefohn, C. Hansen, and E. Praun. Gaussian transfer functions for multi-field volume visualization. *Proceedings of the 14th IEEE Visualization 2003*, pages 65–72, 2003.

[KR70] L. Knopoff and M. J. Randall. The compensated linear-vector dipole: a possible mechanism for deep earthquakes. *Journal of Geophysical Research*, 75(26):4957–4963, 1970.

[KR02] É. Kaminski and N. M. Ribe. Timescales for the evolution of seismic anisotropy in mantle flow. *Geochem. Geophys. Geosyst.*, 158:744–752, 2002.

[KS87] A. Kaufman and E. Shimony. 3d scan-conversion algorithms for voxel-based graphics. In *I3D '86: Proceedings of the 1986 workshop on Interactive 3D graphics*, pages 45–75, New York, NY, USA, 1987. ACM.

[KTW06] G. Kindlmann, X. Tricoche, and C.-F. Westin. Anisotropy creases delineate white matter structure in diffusion tensor mri. In *Proc. of the Ninth Int. Conf. on Medical Image Computing and Computer-Assisted Intervention*, pages 126–133, 2006.

[KvD93] J. J. Koenderink and A. J. van Doorn. Local features of smooth shapes: ridges and courses. *Geometric Methods in Computer Vision II (Proc. of SPIE)*, 2031:2–13, 1993.

[LAG01] A. Leclercq, S. Akkouche, and E. Galin. Mixing triangle meshes and implicit surfaces in character animation. In *Proc. of Eurographic Workshop on Computer animation and simulation*, pages 37–47, 2001.

[LC87] W. E. Lorensen and H. E. Cline. Marching cubes: A high resolution 3d surface construction algorithm. *SIGGRAPH Comput. Graph.*, 21(4):163–169, 1987.

[LCM07] C. Lauterbach, A. Chandak, and D. Manocha. Interactive sound propagation in dynamic scenes using frustum tracing. *IEEE TVCG (Proc. of IEEE Visualization)*, 13(6):1672–1679, 2007.

[Lev88] M. Levoy. Display of surfaces from volume data. *IEEE Comput. Graph. Appl.*, 8(3):29–37, 1988.

[Lev90] M. Levoy. Efficient ray tracing of volume data. *ACM Trans. Graph.*, 9(3):245–261, 1990.

[Lev98] D. Levin. The approximation power of moving least-squares. *Mathematics of Computation*, 67:1517–1531, 1998.

[LGALW09] R. Luis-García, C. Alberola-López, and C. F. Westin. Segmentation of Tensor Fields: Recent Advances and Perspectives. *Tensors in Image Processing and Computer Vision*, pages 35–58, 2009.

[LGS99] T. M. Lehmann, C. Gönner, and K. Spitzer. Survey: Interpolation methods in medical image processing. *IEEE Transactions on Medical Imaging*, 18:1049–1075, 1999.

[LHL$^+$08] J. Liu, W. T. Hewitt, W. R. B. Lionheart, J. Montaldi, and M. Turner. A lemon is not a monstar: visualization of singularities of symmetric second rank tensor fields in the plane. *Eurographics UK Theory and Practice of Computer Graphics*, pages 99–106, 2008.

[Lin96] T. Lindeberg. Edge detection and ridge detection with automatic scale selection. *International Journal of Computer Vision*, 30:465–470, 1996.

[LL94] P. Lacroute and M. Levoy. Fast volume rendering using a shear-warp factorization of the viewing transformation. In *SIGGRAPH '94: Proceedings of the 21st annual conference on Computer graphics and interactive techniques*, pages 451–458, 1994.

[LL99] W. De Leeuw and R. Van Liere. Visualization of global flow structures using multiple levels of topology. In *Data Visualization*, pages 45–52, 1999.

[LL00] Z.-P. Liang and P. C. Lauterbur. *Principles of Magnetic Resonance Imaging: A Signal Processing Perspective.* SPIE Optical Engineering Press, 2000.

[Lla07] I. Llamas. Real-time voxelization of triangle meshes on the gpu. In *SIGGRAPH '07: ACM SIGGRAPH 2007 sketches*, 2007.

[LLP$^+$10] R. Li, L. Liu, L. Phan, S. Abeysinghe, C. Grimm, and T. Ju. Polygonizing extremal surfaces with manifold guarantees. In *SPM '10: Proceedings of the 14th ACM Symposium on Solid and Physical Modeling*, pages 189–194, 2010.

[LN06] T. Lokki and V. Nenonen. Immersive visualization of room acoustics. In *Joint Baltic-Nordic Acoustics Meeting*, Gothenburg, Sweden, November 8–10 2006.

[LRK10] M. Lai, D. Rubin, and E. Krempl. *Introduction to Continuum Mechanics.* Elsevier, 2010.

[MG91] L. B. Montefusco and C. Guerrini. A domain decomposition method for scattered data approximation on a distributed memory multiprocessor. In *EDMCC2: Proc. Dist. Memory Computing*, pages 274–282, 1991.

[MGW05] M. D. Meyer, P. Georgel, and R. T. Whitaker. Robust particle systems for curvature dependent sampling of implicit surfaces. In *Proceedings of the International Conference on Shape Modeling and Applications 2005*, pages 124–133, 2005.

[Mil63] J. Milnor. *Morse Theory.* Princeton University Press, 1963.

[MLPK01] J. Merimaa, T. Lokki, T. Peltonen, and M. Karjalainen. Measurements, analysis, and visualization of directional room responses. In *Proceedings of the 111th Audio Engineering Society (AES) Convention*, New York, NY, USA, September 21–24 2001.

[MMH04] N. Muller, L. Magaia, and B. M. Herbst. Singular value decomposition, eigenfaces, and 3d reconstructions. *Society for Industrial and Applied Mathematics, SIAM REVIEW*, 46(3):518–545, 2004.

[MNKW07] M. Meyer, B. Nelson, R. M. Kirby, and R. Whitaker. Particle systems for efficient and accurate high-order finite element visualization. *IEEE Transactions on Visualization and Computer Graphics*, 13(5):1015 –1026, 2007.

[MOD00] M. Monks, B.M. Oh, and J. Dorsey. Audioptimization: Goal-based acoustic design. *IEEE Computer Graphics and Applications*, 20(3):76–91, 2000.

[MS92] H. P. Moreton and C. H. Sequin. Functional optimization for fair surface design. *Computer Graphics (Proc. of SIGGRAPH '92)*, 26(2):167–176, 1992.

[MSM96] J. G. Moore, S. A. Schorn, and J. Moore. Methods of classical mechanics applied to turbulence stresses in a tip leakage vortex. *Journal of Turbomachinery*, 118(4):622–629, 1996.

[MVdF03] B. Mederos, L. Velho, and L. H. de Figueiredo. Moving least squares multiresolution surface approximation. In *Proceedings of the Computer Graphics and Image Processing 2003*, pages 19–26, 2003.

[MvKK02] A. K. McNamara, P. E. van Keken, and S.-I. Karato. Development of anisotropic structure in the earth's lower mantle by solid-state convection. *Nature*, 416:310–314, 2002.

[Nea04] A. Nealen. An as-short-as-possible introduction to the least squares, weighted least squares and moving least squares methods for scattered data approximation and interpolation. Technical Report, Discrete Geometric Modeling Group, TU Darmstadt, 2004.

[NJP05] A. Neeman, B. Jeremic, and A. Pang. Visualizing tensor fields in geomechanics. In *IEEE Visualization*, pages 35–42. IEEE Computer Society, 2005.

[Obe08] H. Obermaier. Feature-based visualization of grid-less vector fields. *Diploma Thesis. University of Kaiserslautern*, 2008.

[OBH+] H. Obermaier, M. I. Billen, H. Hagen, M. Hering-Bertram, and B. Hamann. Visualizing strain anisotropy in mantle flow fields. *Comput. Graph. Forum (submitted).*

[OBHHB11] H. Obermaier, M. I. Billen, H. Hagen, and M. Hering-Bertram. Interactive visualization of scattered moment tensor data. In *Visualization and Data Analysis 2011, Proc. of SPIE-IS&T Electronic Imaging, SPIE Vol. 7868*, 2011.

[OBS04] Y. Ohtake, A. Belyaev, and H.-P. Seidel. Ridge-valley lines on meshes via implicit surface fitting. In *SIGGRAPH '04: ACM SIGGRAPH 2004 Papers*, pages 609–612, 2004.

[OHBH10] H. Obermaier, M. Hering-Bertram, and H. Hagen. Time-surface maps. *Abstract: VisWeek 2010 Workshop on Foundations of Topological Analysis*, 2010.

[OHBKH09a] H. Obermaier, M. Hering-Bertram, J. Kuhnert, and H. Hagen. Generation of adaptive streak surfaces using moving least squares. In *Proceedings of Dagstuhl 2009, Scientific Visualization Seminar (preprint)*, 2009.

[OHBKH09b] H. Obermaier, M. Hering-Bertram, J. Kuhnert, and H. Hagen. Volume deformations in grid-less flow simulations. *Comput. Graph. Forum*, 28(3):879–886, 2009.

[OHBKH11] H. Obermaier, M. Hering-Bertram, J. Kuhnert, and H. Hagen. On moving least squares based flow visualization. *OpenAccess Series in Informatics (preprint)*, 2011.

[OKHBH09] H. Obermaier, J. Kuhnert, M. Hering-Bertram, and H. Hagen. Stream volume segmentation of grid-less flow simulation. In *Topological Methods in Data Analysis and Visualization: Theory, Algorithms and Applications (Proc. of TopoInVis 2009)*, pages 127–138, 2009.

[ÖM03] E. Özarslan and T. H. Mareci. Generalized diffusion tensor imaging and analytical relationships between diffusion tensor imaging and high angular resolution diffusion imaging. *Magnetic Resonance in Medicine*, 50:955–965, 2003.

[OMD$^+$] H. Obermaier, J. Mohring, E. Deines, M. Hering-Bertram, and H. Hagen. On mesh-free valley surface extraction with application to low frequency sound simulation. *IEEE Transactions on Visualization and Computer Graphics (submitted).*

[OS04] M. Ohtsu and M. Shigeishi. Theory and application of moment tensor analysis in AE. *Proceedings of the Third International Conference on Emerging Technologies in NDT*, pages 19–26, 2004.

[OU04] A. Omoto and H. Uchida. Evaluation method of artificial acoustical environment: Visualization of sound intensity. *Journal of Physiological Anthropology and Applied Human Science*, 23:249–253, 2004.

[PL03] V. Pulkki and T. Lokki. Visualization of edge diffraction. *Acoustics Research Letters Online (ARLO)*, 4(4):118–123, 2003.

[PM90] P. Perona and J. Malik. Scale-space and edge detection using anisotropic diffusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12:629–639, 1990.

[PR99] R. Peikert and M. Roth. The parallel vectors operator - a vector field visualization primitive. In *IEEE Visualization Proceedings*, pages 263–270, 1999.

[PR05a] S. Petrausch and R. Rabenstein. Efficient 3d simulation of wave propagation with the functional transformation method. In *18th Symposium of Simulation Technique, ASIM*, pages 323–330, Erlangen, Germany, September 2005.

[PR05b] S. Petrausch and R. Rabenstein. Highly efficient simulation and visualization of acoustic wave fields with the functional transformation method. In *Simulation and Visualization*, pages 279–290, Otto von Guericke Universität, Magdeburg, March 2005.

[PR05c] S. Petrausch and R. Rabenstein. Simulation of room acoustics via block-based physical modeling with the functional transformation method. In *Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, pages 195–198, New Paltz, New York, October 2005.

[PS08] R. Peikert and F. Sadlo. Height ridge computation and filtering for visualization. In *Proceedings of IEEE Pacific Visualization Symposium*, pages 119–126, 2008.

[PSB10] O. Pasternak, N. Sochen, and P. J. Basser. The effect of metric selection on the analysis of diffusion tensor mri data. *NeuroImage*, 49(3):2190–2204, 2010.

[PvWPS95] F.J. Post, T. van Walsum, F.H. Post, and D. Silver. Iconic techniques for feature visualization. In *Proc. of IEEE Conference on Visualization, 1995*, pages 288–295, 464, 1995.

[RJF$^+$09] P. R. Rodrigues, A. Jalba, P. Fillard, A. Vilanova, and B. M. ter HaarRomeny. A multi-resolution watershed-based approach for the segmentation of diffusion tensor images. *MICCAI Workshop on Diffusion Modelling*, pages 161–172, 2009.

[RL08] P. Rosenthal and L. Linsen. Smooth surface extraction from unstructured point-based volume data using PDEs. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1531–1546, 2008.

[RM00] J. B. T. M. Roerdink and A. Meijster. The watershed transform: Definitions, algorithms and parallelization strategies. *Fundamenta Informaticae*, 41:187–228, 2000.

[RNM09] N. Raghuvanshi, R. Narai, and L. C. Ming. Efficient and accurate sound propagation using adaptive rectangular decomposition. *IEEE TVCG*, 15(5):789–801, September/October 2009.

[Ros00] H. E. Rose. *Linear algebra: a pure mathematical approach.* Birkhäuser Verlag, Bern, Switzerland, 2000.

[RS94] R. M. Russo and P. G. Silver. Trench-parallel flow beneath the nazca plate from seismic anisotropy. *Science*, 263(5150):1105–1111, 1994.

[RS05] M. Reid and B. Szendöi. *Geometry and Topology.* Cambridge University Press, 2005.

[RSP07] T. Rohlfing, E. V. Sullivan, and A. Pfefferbaum. Divergence-based framework for diffusion tensor clustering, interpolation, and regularization. In *Proceedings of the 20th international conference on Information processing in medical imaging*, pages 507–518. Springer-Verlag, 2007.

[RTF$^+$06] F. Rössler, E. Tejada, T. Fangmeier, T. Ertl, and M. Knauff. Gpu-based multi-volume rendering for the visualization of functional brain images. In *In Proc. of SimVis 2006*, pages 305–318, 2006.

[Sab22] W. C. Sabine. *Collected Papers on Acoustics*. Cambhridge: Harvard University Press, 1922.

[SG89] A. Stettner and D. P. Greenberg. Computer graphics visualization for acoustic simulation. In *International Conference on Computer Graphics and Interactive Techniques*, pages 195–206. ACM, 1989.

[SK10] T. Schultz and G. L. Kindlmann. Superquadric glyphs for symmetric second-order tensors. *Visualization and Computer Graphics, IEEE Transactions on*, 16(6):1595 –1604, 2010.

[SMA00] A. Sanna, B. Montrucchio, and R. Arina. Visualizing unsteady flows by adaptive streaklines. *Proc. WSCG 2000*, pages 84–91, 2000.

[SP07] F. Sadlo and R. Peikert. Efficient visualization of lagrangian coherent structures by filtered amr ridge extraction. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1456–1463, 2007.

[STS07] T. Schultz, H. Theisel, and H.-P. Seidel. Segmentation of DT-MRI anisotropy isosurface. In *Proc. EuroVis 2007*, pages 187–194, 2007.

[STS10] T. Schultz, H. Theisel, and H.-P. Seidel. Crease surfaces: From theory to extraction and application to diffusion tensor mri. *IEEE Transactions on Visualization and Computer Graphics*, 16(1):109–119, January 2010.

[STWE07] T. Schafhitzel, E. Tejada, D. Weiskopf, and T. Ertl. Point-based stream surfaces and path surfaces. In *Graphics Interface 2007*, pages 289–296, 2007.

[SWTH07] J. Sahner, T. Weinkauf, N. Teuber, and H.-C. Hege. Vortex and strain skeletons in eulerian and lagrangian frames. *IEEE Transactions on Visualization and Computer Graphics*, 13(5):980–990, 2007.

[TGE97] C. Teitzel, R. Grosso, and T. Ertl. Efficient and reliable integration methods for particle tracing in unsteady flows on discrete meshes. In *Visualization in Scientific Computing '97*, pages 31–41. Springer, 1997.

[TK02] S. Tiwari and J. Kuhnert. Finite pointset method based on the projection method for simulations of the incompressible navier-stokes equations. *Springer LNCSE: Meshfree Methods for Partial Differential Equations*, 26:373–388, 2002.

[TR03] L. Trautmann and R. Rabenstein. *Digital Sound Synthesis by Physical Modeling using Functional Transformation Models.* Kluwer Academic Publishers, New York, 2003.

[TS03] H. Theisel and H.-P. Seidel. Feature flow fields. In *Proceedings of the symposium on Data visualisation 2003*, VISSYM '03, pages 141–148, 2003.

[TSH01a] X. Tricoche, G. Scheuermann, and H. Hagen. Continuous topology simplification of planar vector fields. In *Proceedings of the conference on Visualization '01*, VIS '01, pages 159–166, 2001.

[TSH01b] X. Tricoche, G. Scheuermann, and H. Hagen. Topology-based visualization of time-dependent 2d vector fields. In *In Data Visualization 2001. Proc. VisSym 01*, pages 117–126, 2001.

[TY05] Y. Tokita and Y. Yamasaki. Visualization of 3-dimensional sound fields by numerical solutions of particle displacement. *Acoust. Sci. & Tech.*, 26(2):215–217, 2005.

[UH91] J. K. Udupa and G. T. Herman, editors. *3D imaging in medicine.* CRC Press, Inc., Boca Raton, FL, USA, 1991.

[UI08] T. Urness and V. Interrante. Streamline visualization of multiple 2d vector fields. *Proceedings of SPIE-IS&T Electronic Imaging, SPIE Vol. 6809-9, VDA 2008*, 2008.

[UIL$^+$06] T. Urness, V. Interrante, E. Longmire, I. Marusic, S. O'Neill, and T. W. Jones. Strategies for the visualization of multiple 2d vector fields. *IEEE Comput. Graph. Appl.*, 26:74–82, July 2006.

[USM96] S.-K. Ueng, C. Sikorski, and K.-L. Ma. Efficient streamline, streamribbon, and streamtube constructions on unstructured grids. *IEEE TVCG*, 2(2):100–110, 1996.

[vdGW99] E. van der Giessen and T. Y. Wu. *Advances in Applied Mechanics.* Academic Press, 1999.

[vFWTS08] W. von Funck, T. Weinkauf, H. Theisel, and H.-P. Seidel. Smoke surfaces: An interactive flow visualization technique inspired by real-world flow experiments. In *IEEE TVCG 2008*, pages 1396–1403, 2008.

[vKvOB$^+$97] M. van Kreveld, R. van Oostrum, C. Bajaj, V. Pascucci, and D. Schikore. Contour trees and small seed sets for isosurface traversal. In *SCG '97: Proceedings of the thirteenth annual symposium on Computational geometry*, pages 212–220, 1997.

[WEE02] D. Weiskopf, K. Engel, and T. Ertl. Volume clipping via per-fragment operations in texture-based volume visualization. In *VIS '02: Proceedings of the conference on Visualization '02*, pages 93–100, 2002.

[Wei04] D. Weiskopf. Dye advection without the blur: A level-set approach for texture-based visualization of unsteady flow. In *Proceedings of Eurographics 2004*, pages 479–488, 2004.

[Wen04] H. Wendland. *Scattered Data Approximation.* Cambridge University Press, 2004.

[Wey05a] S. Weyna. Application of "microflown" probe to visualization of acoustic power flow. In *Polish-Scandinavian Structured Conference on Acoustic*, Poznan-Wagrowiec, Poland, September 11–15 2005.

[Wey05b] S. Weyna. Microflown based identification of vortex shadding in the space of real acoustic flow fields. In *Twelfth International Congress on Sound and Vibration*, Lisbon, Portugal, July 11–14 2005.

[WH06] J. Weickert and H. Hagen. *Visualization and processing of tensor fields.* Springer Berlin, 2006.

[Wil93] R. J. Willemann. Cluster analysis of seismic moment tensor orientations. *Geophysical Journal International*, 115(3):617–634, 1993.

[WKL99] D. M. Weinstein, G. L. Kindlmann, and E. C. Lundberg. Tensorlines: Advection-diffusion based propagation through diffusion tensor fields. In *Proceedings of the 10th IEEE Visualization 1999 Conference (VIS '99)*, VISUALIZATION '99, pages 249–253, 1999.

[WS01] T. Wischgoll and G. Scheuermann. Detection and visualization of closed streamlines in planar flows. *IEEE Transactions on Visualization and Computer Graphics*, 7:165–172, April 2001.

[WSMG06] H.-R. Wenk, S. Speziale, A.K. McNamara, and E. J. Garnero. Modeling lower mantle anisotropy development in a subducting slab. *Earth and Planetary Science Letters*, 245:302–314, 2006.

[WTS$^+$07] A. Wiebel, X. Tricoche, D. Schneider, H. Janicke, and G. Scheuermann. Generalized streak lines: Analysis and visualization of boundary induced vortices. *Visualization and Computer Graphics, IEEE Transactions on*, 13(6):1735 –1742, 2007.

[WTS09] A. Wiebel, X. Tricoche, and G. Scheuermann. Extraction of separation manifolds using topological structures in flow cross sections. In *Topology-Based Methods in Visualization II*, pages 31–44, 2009.

[XD08] Y. Xi and Y. Duan. A novel region-growing based iso-surface extraction algorithm. *Computers & Graphics*, 32(6):647–654, December 2008.

[YST02] T. Yokota, S. Sakamoto, and H. Tachibana. Visualization of sound propagation and scattering in rooms. *Acoust. Sci. & Tech.*, 23(1):40–46, 2002.

[ZCEP07] L. Zhang, W. Chen, D. S. Ebert, and Q. Peng. Conservative voxelization. *The Visual Computer*, 23:783–792, 2007.

[ZK95] S. Zhang and S.-I. Karato. Lattice preferred orientation of olivine aggregates deformed in simple shear. *Nature*, 375:774–777, 1995.

[ZMB$^+$03] L. Zhukov, K. Museth, D. Breen, R. Whitaker, and A. H. Barr. Level Set Modeling and Segmentation of DT-MRI Brain Data. *Journal of Electronic Imaging*, 12:125–133, 2003.

[Zom05] A. J. Zomorodian. *Topology for computing.* Cambridge University Press, 2005.

[ZP04] X. Zheng and A. Pang. Topological lines in 3d tensor fields. In *VIS '04: Proceedings of the conference on Visualization '04*, pages 313–320, 2004.

[ZTTZ05] O. C. Zienkiewicz, R. L. Taylor, R. Leroy Taylor, and J. Z. Zhu, editors. *The finite element method: its basis and fundamentals.* Elsevier Butterworth-Heinemann, 2005.

[ZTW06] U. Ziyan, D. Tuch, and C.-F. Westin. Segmentation of Thalamic Nuclei from DTI using Spectral Clustering. In *Ninth International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI'06)*, Lecture Notes in Computer Science 4191, pages 807–814, Copenhagen, Denmark, October 2006.

List of Figures

2.1. Visualization Pipeline 5
2.2. Topology 7
2.3. Scalar Fields 8
2.4. 1D Analysis 10
2.5. X-Ray 10
2.6. Scalar Field Topology 12
2.7. Wind As Vector 13
2.8. Vector Field Topology 15
2.9. Cube Deformation 16
2.10. Tensor Deformation 18

3.1. Lagrangian & Eulerian Perspective 28
3.2. Critical Points 36

4.1. Point Seeding 48
4.2. Curvature Approximation 48
4.3. Seeding And Merging 50
4.4. Degenerate Lines 52
4.5. Gradient Ascent & Descent 54
4.6. Ribbon Triangulation 55
4.7. Simulated Room 57
4.8. Time-Varying Nodes 58
4.9. Surface Accumulation 59
4.10. Visualization Techniques 60
4.11. Merging Triangulation Fronts 60
4.12. Comparison to Scalar Value Thresholding 61
4.13. Comparison 61
4.14. Noisy Valley Surfaces 62
4.15. Interactive Surface Extraction 63
4.16. Level of Detail Parameter Evaluation 64
4.17. Invalid Convergence 65
4.18. Curvature Adaptivity and Ribbon Triangulation 66

5.1. FTLE Illustration 68
5.2. 2D Tensor Deformation 69
5.3. Matrix Multiplication 71

5.4. Path- and Streakline Comparison 72
5.5. Geophysical Data Sets 77
5.6. Slicing Techniques 77
5.7. 2D Line Rasterization 79
5.8. Sphere Deformation 82
5.9. Pseudo Code for Pixel Shader 86
5.10. Multi-Volume Visualization 87
5.11. Geometric Volume Extraction 87
5.12. Sphere Parametrization 88
5.13. Bin Coloring 89
5.14. Ellipsoid Integration 2D 90
5.15. Ellipsoid Integration 3D 91
5.16. Streakline Deformation 91
5.17. Stationary Segmentation I 92
5.18. Stationary Segmentation II 92
5.19. Resolution Comparison 93
5.20. Sinking Slab 94
5.21. Strain Lines 94
5.22. Strain Line Opacity 95
5.23. Prominent Features I 96
5.24. Prominent Features II 97
5.25. Volume Evolution I 98
5.26. Volume Evolution II 98

6.1. Streakline Boundary Fitting 105
6.2. Non Adaptive Streak Surface 107
6.3. Streak Surface Meshing 108
6.4. Integration Order 109
6.5. Ill-Conditioned Triangulation 110
6.6. Tetrahedral Artifacts 111
6.7. MLS Surface Approximation 111
6.8. MAC Streak Surface Color Map 112
6.9. Boundary Edge Splitting 114
6.10. Adaptivity Comparison 115
6.11. Streak Areas 115
6.12. Periodic Streak Areas 116
6.13. Multi-Field Streak Areas 116
6.14. Streak Surface Statistics 118
6.15. 3D Von Kármán Vortex Street 120
6.16. Transparent Surface Comparison 120
6.17. Volume Voxelization 121
6.18. Surface Ellipsoids 122
6.19. Streak Surface Visualization 123

6.20. Refinement Mesh 123
6.21. Surface Mesh Comparison I 124
6.22. Surface Mesh Comparison II 124

7.1. Force Couples and Beachball Representation 127
7.2. Beachball and Glyph Shapes 130
7.3. Glyph Visualization Types 131
7.4. Slip Geometry Glyphs 132
7.5. Fault Geometry in Beachballs 134
7.6. Illustration of Cluster Creation with MST 137
7.7. Stereographic Projection 139
7.8. Cluster Visualization by Watermarking. 142
7.9. Cluster Selection and Outlining 142
7.10. Similarity Querying and Outlining 143
7.11. Visualization of Representative Moment Tensor Glyphs 143
7.12. Comparison of Classic and New Visualization Technique 143

B.1. Voxelization 152
B.2. Volume Voxelization 153

Curriculum Vitae

Personal Details

Name	Harald Obermaier
Date of Birth	26.11.1983
Place of Birth	Konstanz, Germany
Citizenship	Austria

Education

1990–1994	Wollmatinger Grundschule, Konstanz, Germany; Elementary School
1994–2000	Geschwister Scholl Schule, Konstanz, Germany; Secondary School
2000/2001	Lake Braddock High School, Fairfax County, Virginia, USA
2001–2003	Geschwister Scholl Schule, Konstanz, Germany; Secondary School Certificate: High-school diploma (Abitur)
2003– Aug. 2008	Study of Computer Science, University of Kaiserslautern, Germany Degree: Dipl.-Inf. (Diplom-Informatiker) Diploma Thesis: Feature-Based Visualization of Grid-Less Vector Fields.
2008– Feb. 2011	PhD student of the International Research Training Group 1131 at the University of Kaiserslautern, Germany.

Work Experience

2009–2010	Teaching assistant for ”Computer-Animation” and ”Visualization and Virtual Reality” lectures

Sunday 27th February, 2011